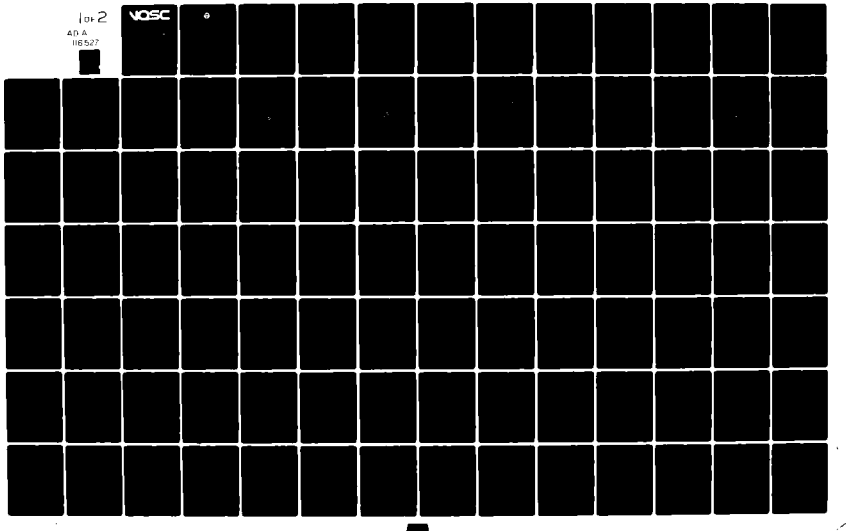
AD-A116 527 SYSTEM DEVELOPMENT CORP SAN DIEGO CA F/G 17/2
STAMMER: SYSTEM FOR TACTICAL ASSESSMENT OF MULTISOURCE MESSAGES--ETC(U)
MAY 79 R J BECHTEL, P H MORRIS N00123-76-C-0172
UNCLASSIFIED NOSC-TD-252 NL



NOSC

NOSC TD 252

NOSC TD 252

Technical Document 252

STAMMER:

system for tactical assessment of multisource messages, even radar

R. J. Bechtel
P. H. Morris
Systems Development Corporation

May 1979

Prepared for
Naval Electronic Systems Command

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

This work was performed by the Tactical Command and Control Division, (Code 824) as a part of the Tactical Situation Assessment (TSA) problem under NOSC Project No. CC18. Work was performed by Mr. Robert J. Bechtel and Dr. Paul H. Morris of Systems Development Corporation under Contract N00123-76-C-0172. This TSA task is a part of Command and Control Block Program (62721N-XF21-201.100) sponsored by NAVELEX, Code 330, the Command and Control Division of the Research and Technology Directorate, NAVELEX, Code 03. A part of the funding for this effort was from The Information Processing Technology Office of DARPA under DARPA Order 3454.

Released by
R.C. Kolb, Head
Tactical Command and Control Division

Under Authority of
J.H. Maynard, Head
Command Control-Electronic Warfare
Systems and Technology Department

A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NOSC TD 252 (TD 252) — 2. GOVT ACCESSION NO.: AD-A116527	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: STAMMER: system for tactical assessment of multisource messages, even radar	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): R. J. Bechtel, P. H. Morris, Systems Development Corporation	8. CONTRACT OR GRANT NUMBER(s): N00123-76-C-0172
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems Development Corporation, 3065 Rosecrans Place, San Diego, CA 92110	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62721N-XF21-201.100
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Electronic Systems Command, Washington, D. C.	12. REPORT DATE: May 1979
	13. NUMBER OF PAGES: 109
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Naval Ocean Systems Center, San Diego, CA	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

STAMMER was developed to serve as a demonstration of the applicability of rule-based inference techniques to the Tactical Situation Assessment (TSA) problem. In addition, it serves as a testbed for development of new inference rules and techniques.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

CONTENTS

1. PURPOSE

STAMMER was developed to serve as a demonstration of the applicability of rule-based inference techniques to the Tactical Situation Assessment (TSA) problem. In addition, it serves as a testbed for development of new inference rules and techniques. To achieve these goals, the following features were included in STAMMER:

1) **A graphics interface.** STAMMER uses a slightly modified version of DSPLA (Ref. 1) to display platform tracks and to provide a visual presentation of position information.

2) **Explanation of inference.** Whenever the system uses a rule to reach some conclusion, the user may examine the data base, rules, and derivations to discover why the conclusion is considered valid.

3) **Ease of rule addition.** A primary reason for adopting a rule-based inference approach is the flexibility and ease of modification that inference rules provide. A convenient mechanism for adding rules makes this flexibility available.

4) **A flexible data base structure.** While the structure of the demonstration data base is fixed, functions are provided to allow redefinition of the data base structure for different applications.

5) **Multiple terminal configurations.** The ideal terminal arrangement for STAMMER includes both a controlling text terminal and a slaved graphic display terminal. However, such a configuration is not always available, so STAMMER has been designed to run in standalone text and standalone graphics modes as well as a two-terminal mode.

6) **Demonstration scenario.** To make demonstration simple, and to allow concentration on a working system, a scenario, consisting of data base structure definition, a technical data base, production rules, and messages has been provided. Additionally, an executable file containing the scenario has been provided.

7) **Measures of confidence.** Realizing that many useful rules are not absolute, STAMMER works with confidences rather than logical truth values in its inferences. These confidences are roughly related to probabilities and provide a broad range of measures for the strength of any conclusion.

II. SYSTEM OVERVIEW

STAMMER serves as an organizer of information. It collects information by receiving messages and radar reports, and organizes this raw data into graphic displays and textual commentary to aid in tactical situation assessment. The organization that STAMMER performs is more than simple formatting, however. Through the use of specified rules, the system combines information from messages and reports to draw conclusions about the situation in the vicinity of the home ship. These conclusions are reflected in both the display and commentary. The system data base is available for examination, and includes not only the original raw data, but also information about why and how the conclusions were reached.

During execution, the user will see the following cycle repeated as long as messages and reports are received:

(1) A message or report is received -- the user is informed, and the critical information in the message is printed for reference.

(2) A display, showing the area situation with the new information, is drawn. The user may manipulate this image.

(3) The system makes some commentary on the conclusions it can reach, on the basis of the new information.

(4) If any conclusions were reached, the user is given the opportunity to query the system about the contents of its data base.

III. DEMONSTRATION SCENARIO AND SELECTED EXAMPLES

In this demonstration run we have used the following scenario: the home ship *Perry* is stationary off the coast of Iceland with radar on. A storm covers the northwest portion of the island and the surrounding ocean area. The *Perry* begins to move towards St. Johns, roughly paralleling a merchant lane, so that ships within the lane will fall inside its radar coverage. Merchant3 is travelling in the lane from St. Johns to Reykjavik. Red and Blue are hostile warships with the intention of intercepting Merchant3. They attempt to simulate merchants by travelling at merchant speed in two other lanes. At a convenient point they will make a sudden dash at maximum speed to reach the lane of interest.

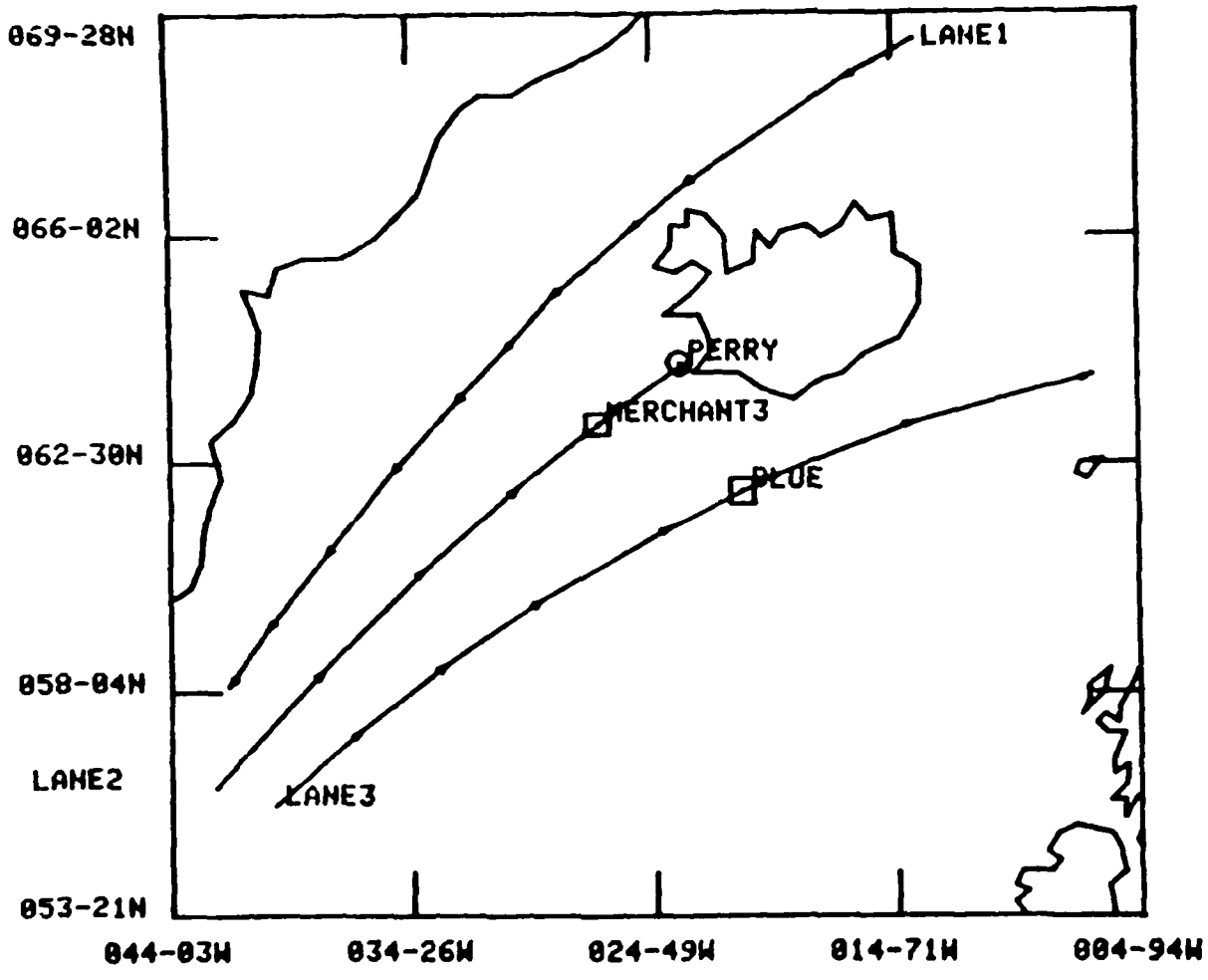
Patrol17 flies from Reykjavik to the southern tip of Greenland, reporting on platforms in the area. The *Perry* receives information from it, from ground-based stations in the region, and from its own radar. The system demonstrated is supposed to be onboard the *Perry*, collating this information.

It should be stressed that the system has no knowledge of the identities of platforms or other aspects of the scenario, other than that which it receives in messages or deduces from the information available to it.

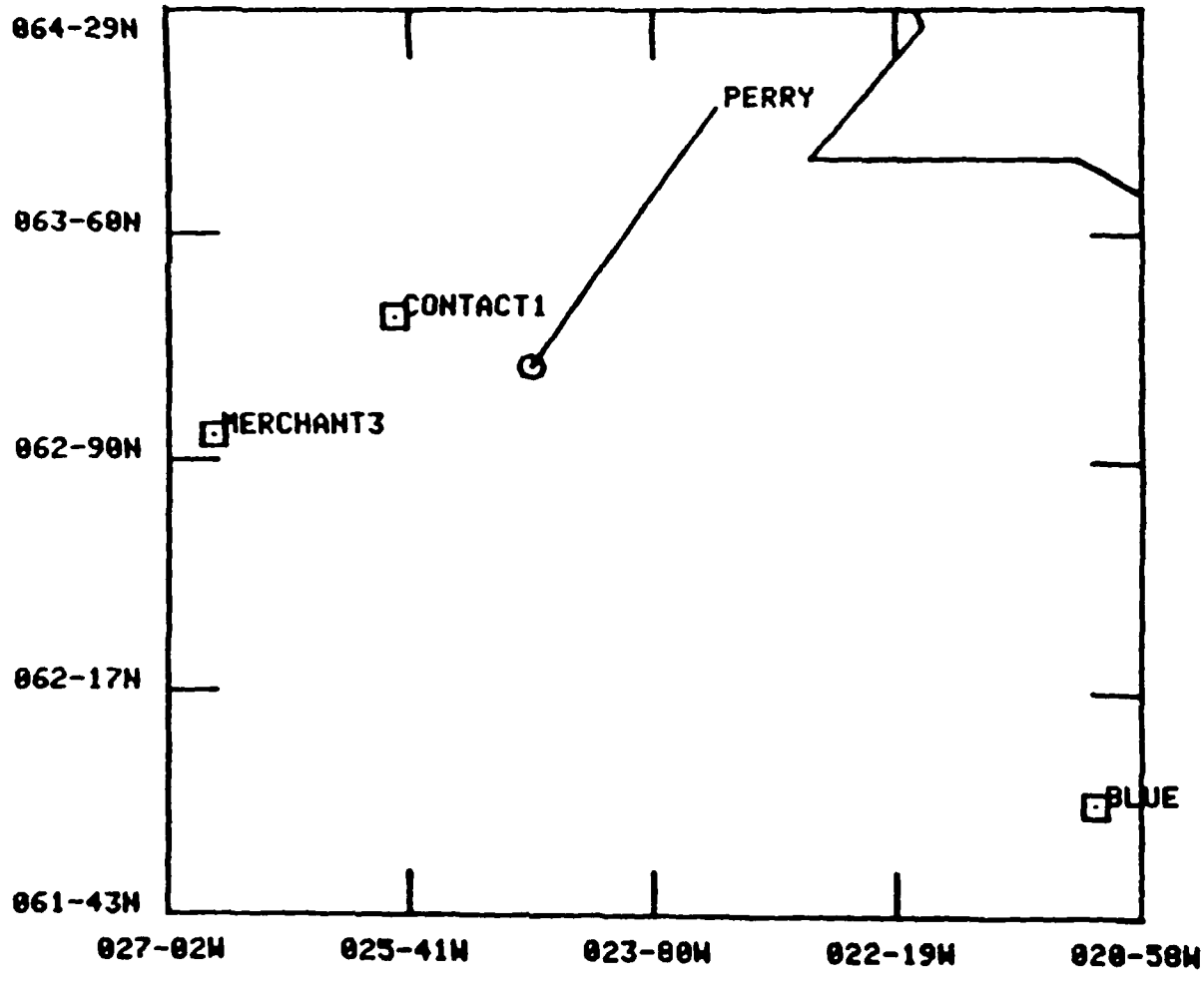
Comments added to the typescript are preceded by a semi-colon. Lines consisting of three asterisks indicate sections omitted for brevity.

```
>EXECUTE

Are you running on a Tektronix?...No
Do you have a Tektronix available for display? No


 *** WEATHER REPORT RECEIVED.

Time: 0
Storm centered at 65.83, -24.45,  approximate diameter 175.7003

No rules were satisfied.          ;no deductions triggered at this point.

  ***MESSAGE RECEIVED. Message number 1

Time: 360
Concerns a new platform. Assigned working name MERCHANT3
Content:

  (CATEGORY: MERCHANT SPEED: 10 LATITUDE: 62.97 LONGITUDE: -26.73 TOS: 180
             SOURCE: EXTERNAL)  ;the message identifies the platform
                                ;as a merchant.

1. NON-COMBATANT is the WARLIKE of MERCHANT3     ;deductions from the
                                                 ;technical database,
2. CIVILIAN is the TYPE of MERCHANT3             ;since it is known
                                                 ;the platform is a
3. SURFACE is the MEDIUM of MERCHANT3            ;merchant.

4. COMBATANT is not the WARLIKE of MERCHANT3

5. AIR is not the MEDIUM of MERCHANT3

6. SIGHTING0341 is INSIDE-A-MERCHANTLANE

Explanation system
Type HELP for help, CR to exit.
QUESTION ?
Leaving EXPLAIN.
```

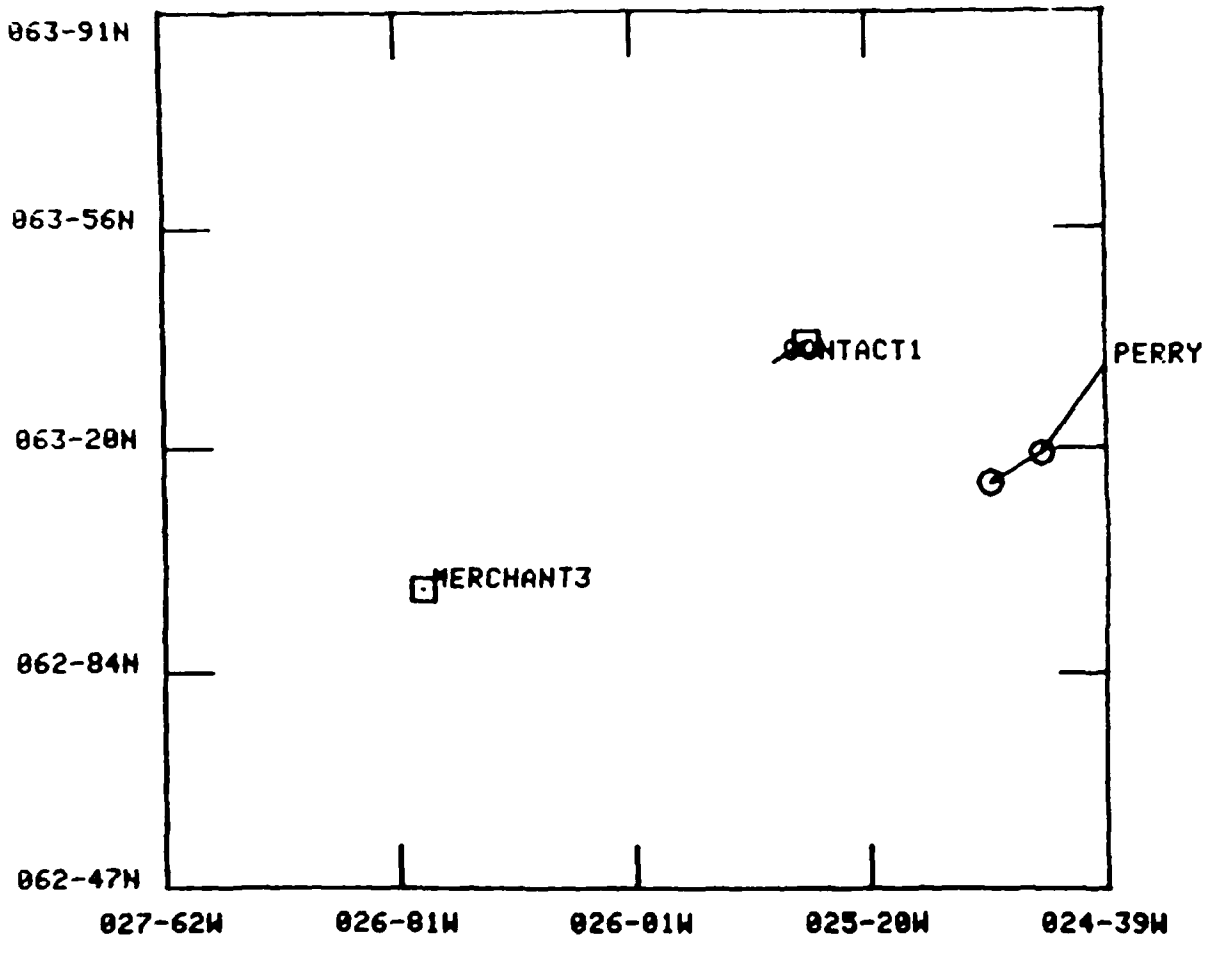
```
  ***MESSAGE RECEIVED. Message number 2

Time: 400
Concerns a new platform. Assigned working name BLUE
Content:

  (TOS: 150 LATITUDE: 61.8 LONGITUDE: -20.87 SOURCE: EXTERNAL)


1. SIGHTING0380 is INSIDE-A-MERCHANTLANE

        ; the system has assigned the name sighting0380 to the sighting
        ; just received and observes that it is in a merchant lane.

Explanation system
Type HELP for help, CR to exit.
QUESTION ? Tell me about MERCHANTLANE


1. LANE1 is a    MERCHANTLANE
2. LANE2 is a    MERCHANTLANE
3. LANE3 is a    MERCHANTLANE

QUESTION ?
Leaving EXPLAIN.
```


069-28N
066-02N
062-30N
058-04N
053-21N
044-03W
034-26W
024-49W
014-71W
004-94W
LANE1
LANE2
LANE3
PERRY
MERCHANT3
BLUE

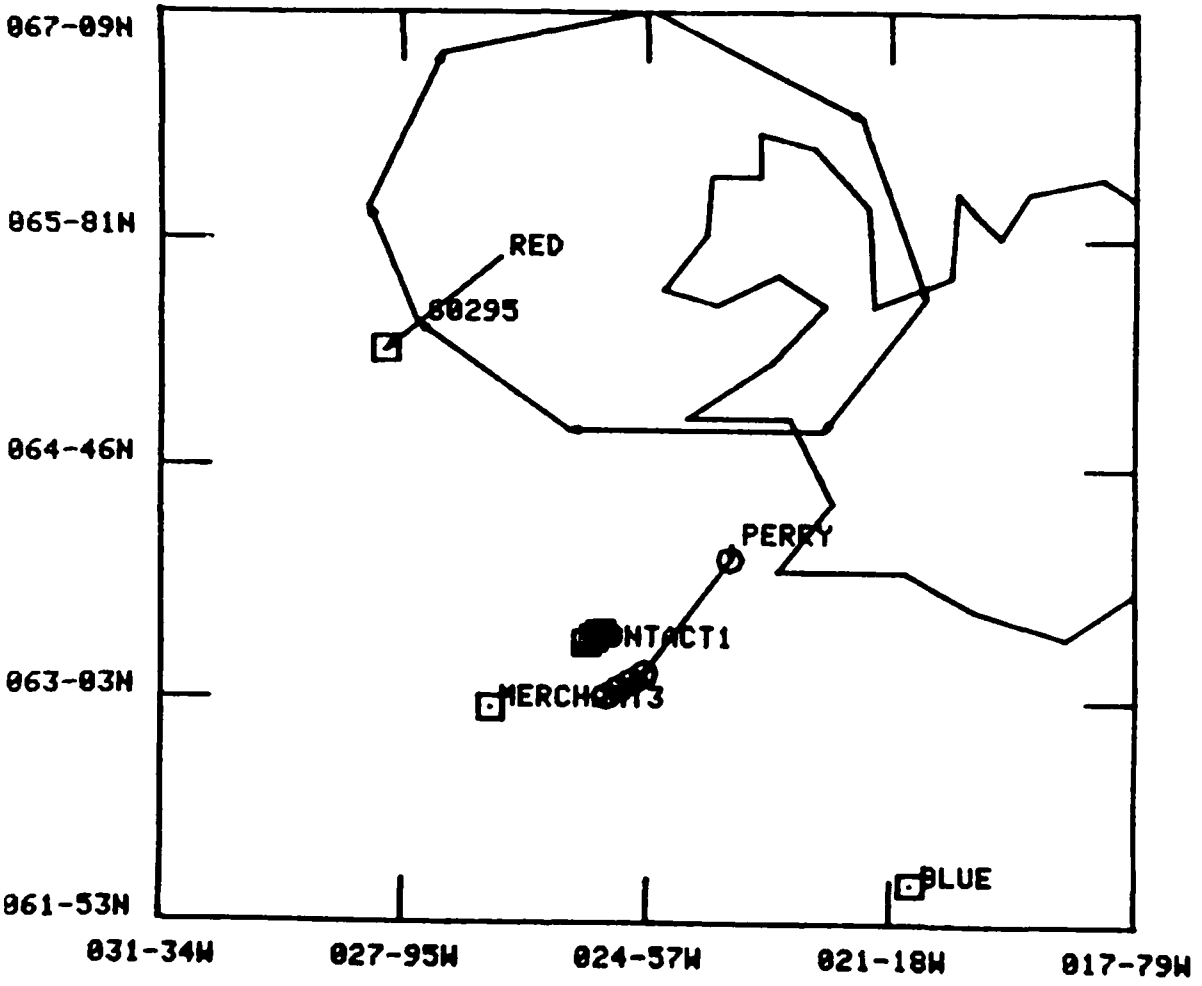
```
  ***RADAR REPORT.

Time: 420
Location: Range: 26.15674  Bearing: 290.5145
Assumed to be a new contact, assigned working name CONTACT1

        ;this is a radar sighting of MERCHANT3, although the system
        ;does not know it.

1. SIGHTING0400 is INSIDE-A-MERCHANTLANE

2. MERCHANT is somewhat likely to be the CATEGORY of CONTACT1
  because no known combatant could have reached it.

3. NON-COMBATANT is somewhat likely to be the WARLIKE of CONTACT1

4. CIVILIAN is somewhat likely to be the TYPE of CONTACT1

5. SURFACE is somewhat likely to be the MEDIUM of CONTACT1

6. COMBATANT is somewhat unlikely to be the WARLIKE of CONTACT1

7. AIR is somewhat unlikely to be the MEDIUM of CONTACT1

        ;since the system has deduced some likelihood of contact1
        ;being a merchant, it extends this likelihood to include
        ;the properties of merchants, as given in the technical
        ;data base.


Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHY (ready to be asked about an assertion) 4


1. ID8  ;a system rule, used to implement the technical data base.

QUESTION ? WHY (ready to be asked about a rule)


1. CONTACT1 is a   PLATFORM
2. MERCHANT is somewhat likely to be the CATEGORY of CONTACT1
3. CIVILIAN is the TTYPE of MERCHANT     ;item in technical data base.

Do you want to know about any of these? No


QUESTION ?
Leaving EXPLAIN.
```


064-29N
063-60N
062-90N
062-17N
061-43N
027-02W
025-41W
023-00W
022-19W
020-58W
PERRY
CONTACT1
MERCHANT3
BLUE

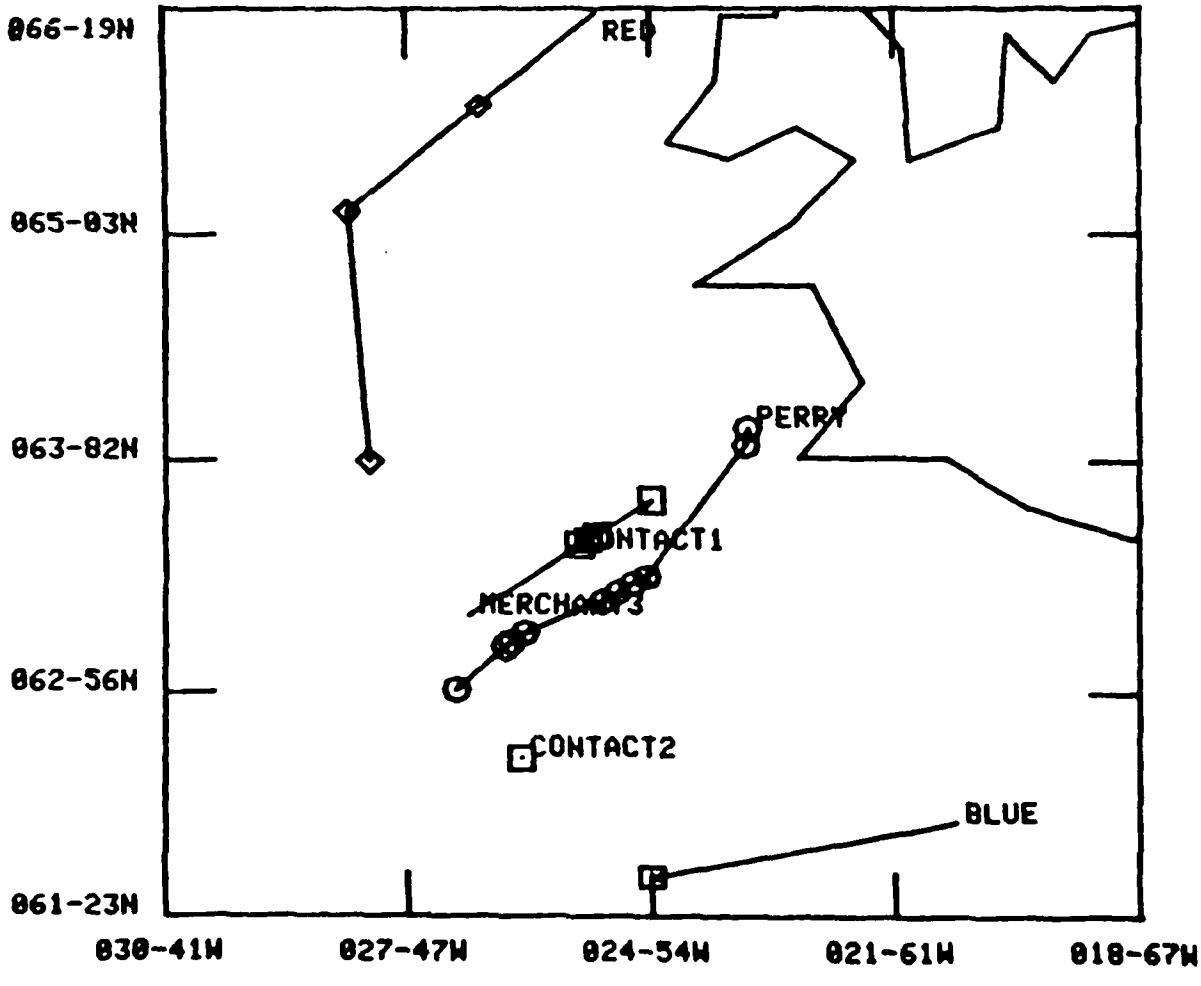
```
  ***RADAR REPORT.

Time: 440
Location: Range: 21.7085  Bearing: 309.0736
Assumed to be CONTACT1

1. MERCHANT is very probably the CATEGORY of CONTACT1
   because its course and speed match those of a known merchant.

        ;the "very probably" indicates the CUMULATIVE effect of the
        ;evidence thus far.

2. SIGHTING0434 is INSIDE-A-MERCHANTLANE                              ;

Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHY (ready to be asked about an assertion) 1

        ;i.e. summarize the evidence for the assertion in 1.

1. A-MATCH-FOR-A-KNOWN-MERCHANT          ;these are the rules
2. NOT-REACHABLE-BY-ANY-COMBATANT        ;that were used.

QUESTION ? WHY (ready to be asked about a rule) 1

1. CONTACT1 is a   CONTACT
2. MERCHANT3 is a   MERCHANT
3. SIGHTING0434 is the LAST-SIGHTING of CONTACT1
4. 63.37 is the LATITUDE of SIGHTING0434
5. -25.41 is the LONGITUDE of SIGHTING0434
6. 55.53572 is the COURSE of SIGHTING0434
7. 10.45755 is the SPEED of SIGHTING0434
8. 440 is the TIME of SIGHTING0434
9. SIGHTING0341 is the LAST-SIGHTING of MERCHANT3
10. 180 is the TIME of SIGHTING0341
11. 62.97 is the LATITUDE of SIGHTING0341
12. -26.73 is the LONGITUDE of SIGHTING0341
13. 55.53572 is ROUGHLY-THE-SAME-COURSE-AS 55.53572
14. 10.45755 is ROUGHLY-THE-SAME-SPEED-AS 9.935611
15. 10.45755 is not LESS-THAN 9
16. 10.45755 is not GREATER-THAN 25

        ;these describe in detail the conditions that caused the
        ;rule to fire (succeed).

Do you want to know about any of these? No

QUESTION ? Is CONTACT1 HOSTILE

not to my knowledge      ;merchants may or may not be hostile.

QUESTION ?
Leaving EXPLAIN.
```


063-91N
063-56N
063-20N
062-84N
062-47N
027-62W
026-81W
026-01W
025-20W
024-39W
CONTACT1
PERRY
MERCHANT3

```
                *       *       *

  ***MESSAGE RECEIVED.  Message number 4

Time:  510
New message concerning RED
Content:

  (TOS: 210 LATITUDE: 65.7 LONGITUDE: -26.61 SOURCE: EXTERNAL)


1.  MERCHANT is somewhat unlikely to be the CATEGORY of RED
    because its track crosses a storm.

2.  SIGHTING0521 is INSIDE-A-MERCHANTLANE

Explanation system
Type HELP for help, CR to exit.
QUESTION ? Tell me about STORM


1.  STORM0295 is a   STORM          ;lists the storms.

QUESTION ? WHERE is S0295

The following is a list of lat-lon pairs defining the edge of the storm.
((64.67 -25.66) (64.67 -22.12) (65.46 -20.68) (66.5 -21.57) (
67.11 -24.48) (66.88 -27.41) (65.99 -28.45) (65.3 -27.75))

          ;the storm would also be plotted if the display were on.


QUESTION ?
Leaving EXPLAIN.
```


067-09N
065-81N
064-46N
063-03N
061-53N
031-34W
027-95W
024-57W
021-18W
017-79W
RED
80295
PERRY
BLUE

```
                *       *       *

  ***MESSAGE RECEIVED. Message number 8

Time: 685
New message concerning RED
Content:

  (LATITUDE: 63.81 LONGITUDE: -27.95 TOS: 615 SOURCE: PATROL17 CLASS:
       KYNDA)
       ;this message identifies RED as a hostile warship, KYNDA class.


                *       *       *


  ***RADAR REPORT.

Time: 695
Location: Range: 32.19589  Bearing: 137.7162
Assumed to be a new contact, assigned working name CONTACT2

       ;this is actually a radar sighting of BLUE.

1. MERCHANT is somewhat unlikely to be the CATEGORY of CONTACT2
  because it popped up outside normal merchant range.

2. MERCHANT is somewhat unlikely to be the CATEGORY of CONTACT2
  because it lies outside all known merchant lanes.

Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHY (ready to be asked about an assertion) 1


1. A-DISTANT-POPUP
2. OUTSIDE-ALL-MERCHANTLANES

QUESTION ? WHY (ready to be asked about a rule) 1


1. CONTACT2 is a   CONTACT
2. SIGHTING0664 is the FIRST-SIGHTING of CONTACT2
3. 32.19589 is the RANGE of SIGHTING0664
4. 32.19589 is GREATER-THAN 30

Do you want to know about any of these? No


QUESTION ?
Leaving EXPLAIN.
```


066-19N
065-03N
063-82N
062-56N
061-23N
030-41W
027-47W
024-54W
021-61W
018-67W
RED
PERRY
CONTACT1
MERCHANT3
CONTACT2
BLUE

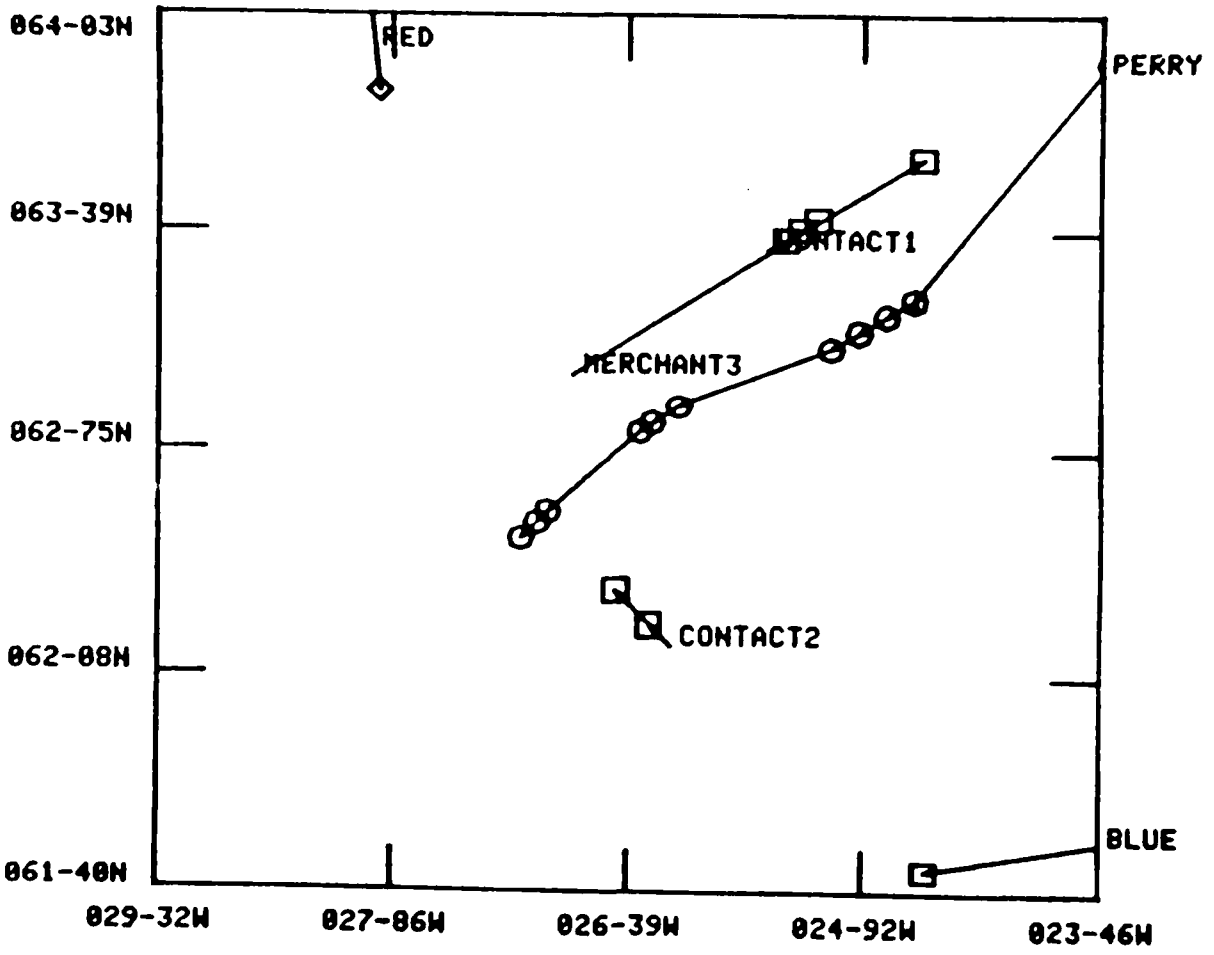
***RADAR REPORT.

Time: 705
Location: Range: 26.82486 Bearing: 133.1563
Assumed to be CONTACT2

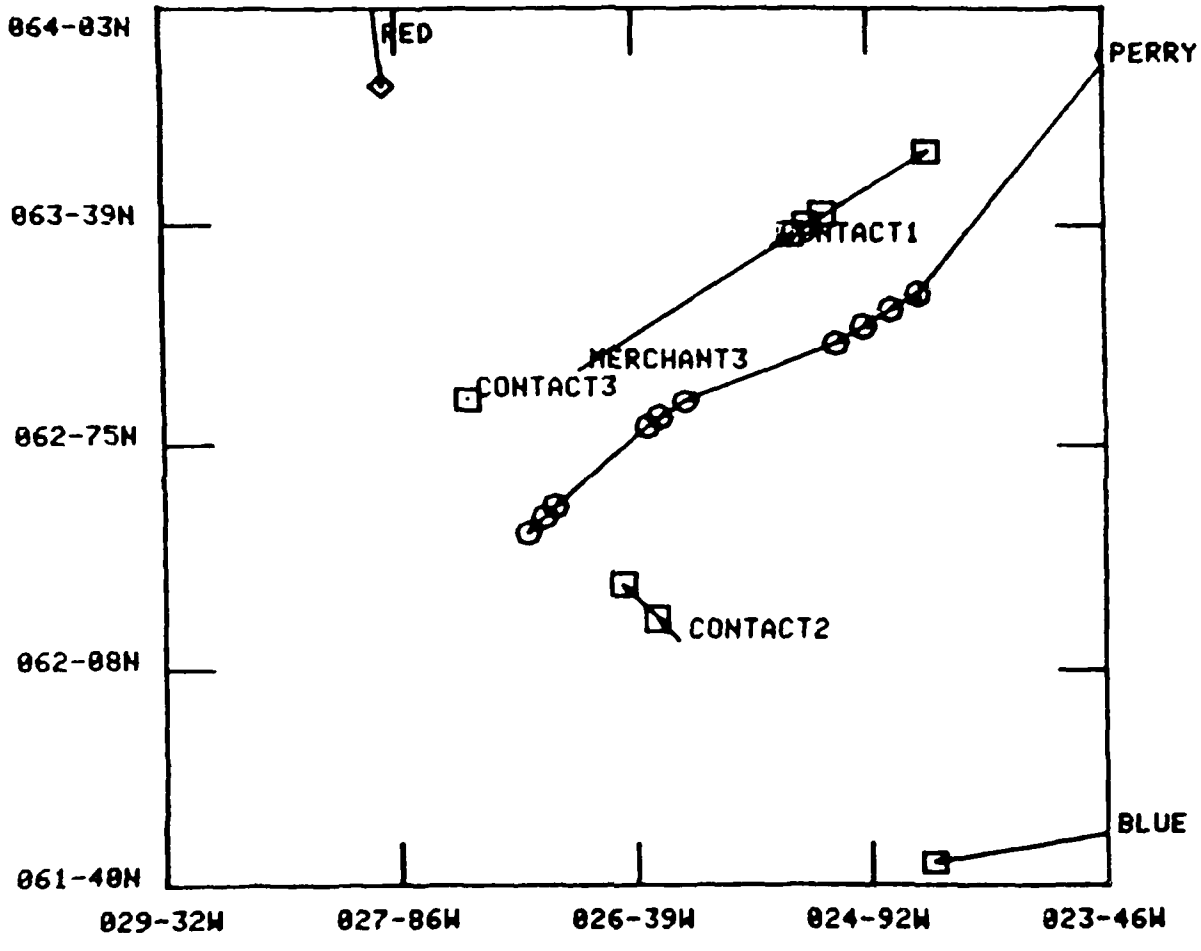
1. MERCHANT is probably not the CATEGORY of CONTACT2
because its speed is greater than maximum merchant cruise speed.

2. MERCHANT is probably not the CATEGORY of CONTACT2
because it lies outside all known merchant lanes.

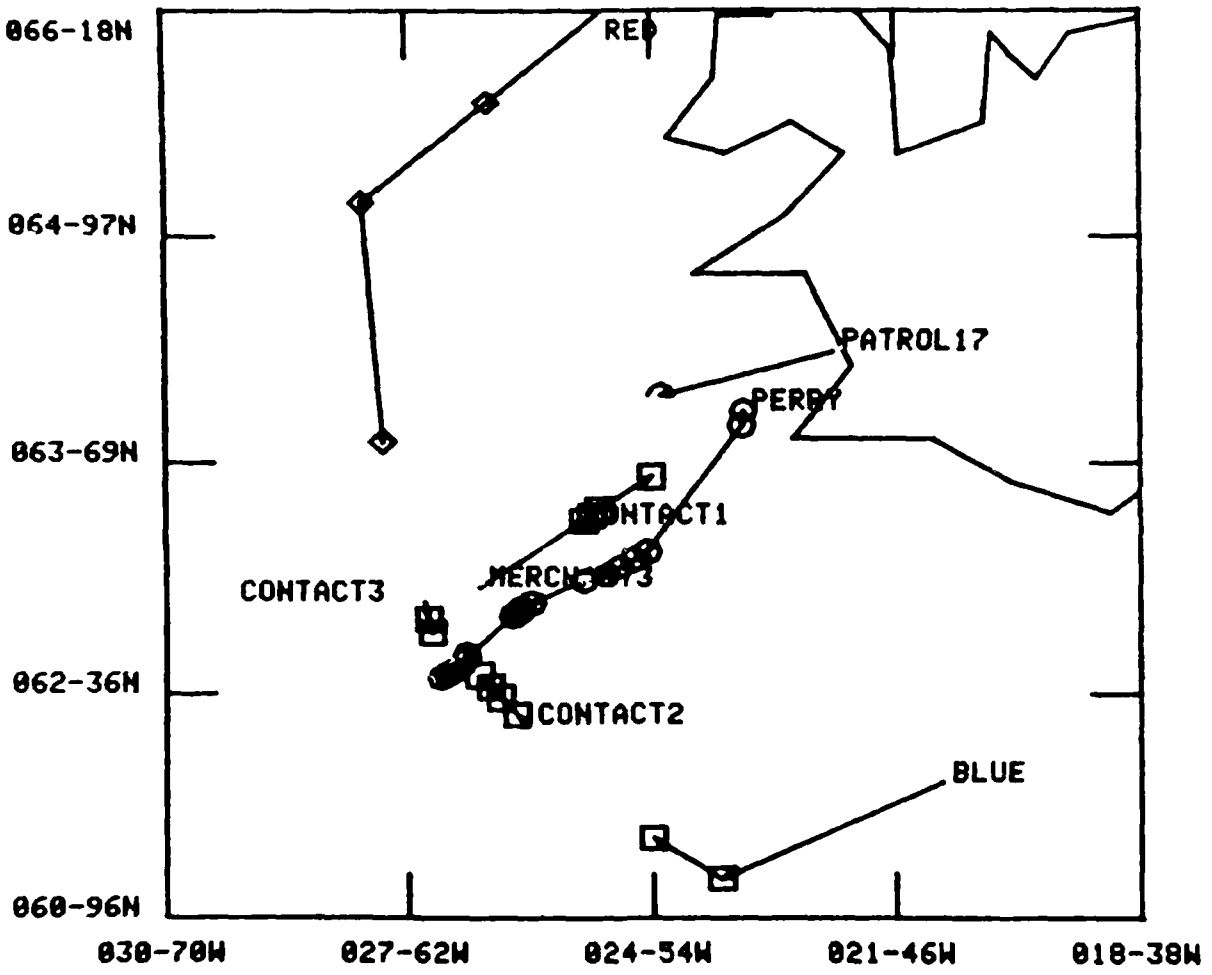
;the evidence mounts that CONTACT2 is not a merchant.


064-03N
063-39N
062-75N
062-08N
061-40N
029-32W
027-86W
026-39W
024-92W
023-46W
RED
PERRY
CONTACT1
MERCHANT3
CONTACT2
BLUE

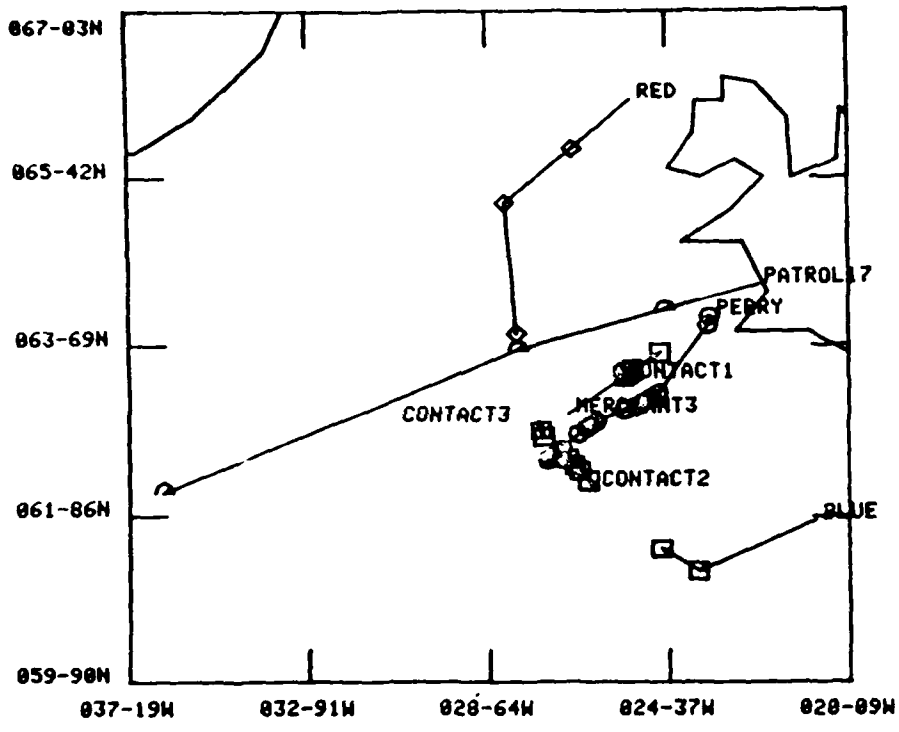
```
                *       *       *

  ***RADAR REPORT.

Time: 720
Location: Range: 25.51945  Bearing: 336.6366
Assumed to be a new contact, assigned working name CONTACT3

        ;contact3 is really a radar sighting of RED.

1. MERCHANT is somewhat unlikely to be the CATEGORY of CONTACT3
   because it lies outside all known merchant lanes.

Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHAT is THE FIRST-SIGHTING OF CONTACT3

1. SIGHTING0737

QUESTION ? Is SIGHTING0737 REACHABLE-BY-A-COMBATANT

yes

QUESTION ? WHOSE WITHIN-REACH  is SIGHTING0737

1. not SIGHTING0562
2. SIGHTING0613

QUESTION ? WHOSE SIGHTING is SIGHTING0562

1. BLUE

QUESTION ? WHOSE SIGHTING is SIGHTING0613

1. RED

        ;although the system is intended only to detect non-merchants,
        ;here the operator has effectively made use of it to suggest the
        ;identity of CONTACT3.

QUESTION ?
Leaving EXPLAIN.
```

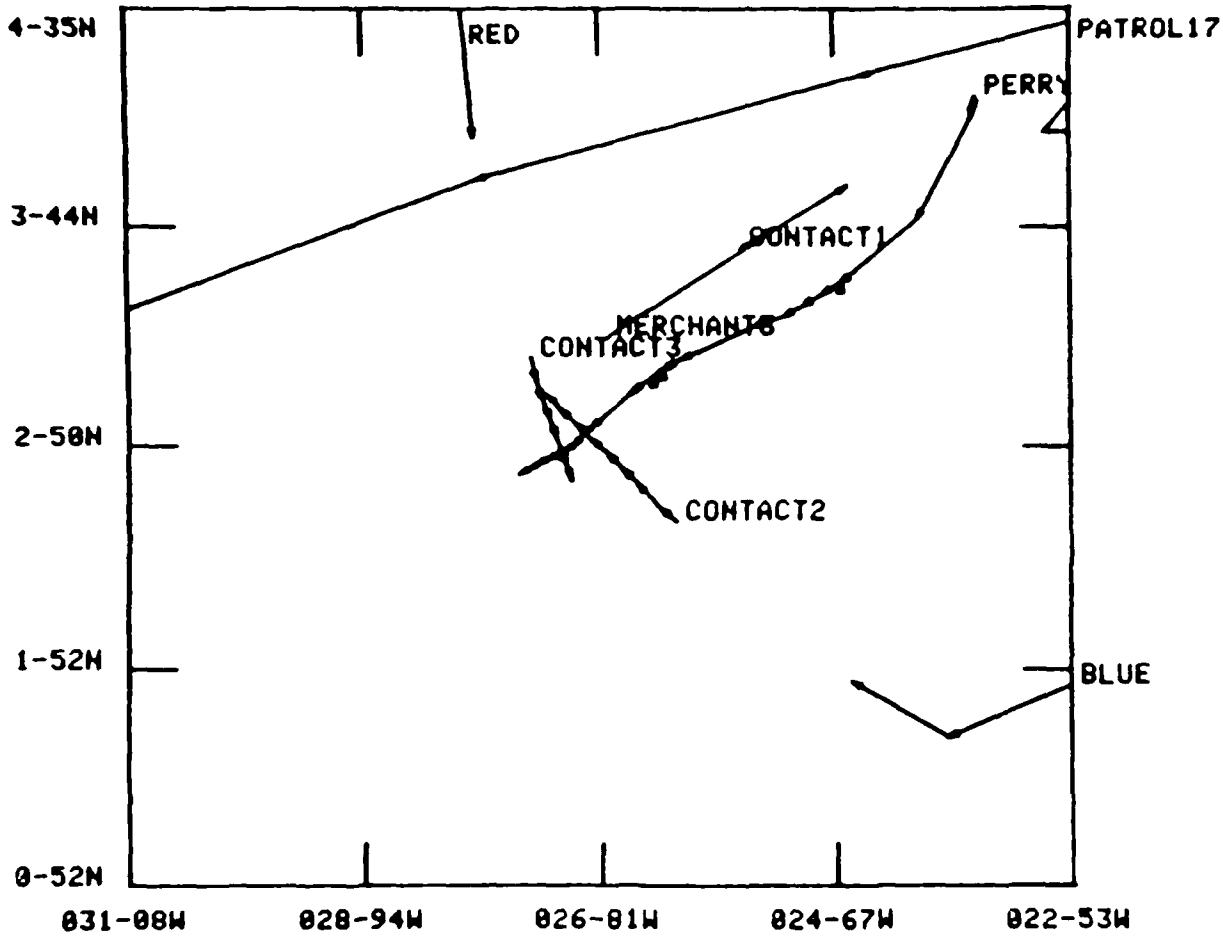

064-03N
063-39N
062-75N
062-08N
061-40N
029-32W
027-86W
026-39W
024-92W
023-46W
RED
PERRY
CONTACT1
MERCHANT3
CONTACT3
CONTACT2
BLUE

```
                *       *       *

  ***MESSAGE RECEIVED. Message number 10

Time: 747
Concerns a new platform. Assigned working name PATROL17
Content:

  (TOS: 600 LATITUDE: 64.33 LONGITUDE: -22.24 SOURCE: EXTERNAL NAME: HUMMER
        CLASS: F16-A CATEGORY: FIGHTER)

                *       *       *

Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHOSE SOURCE is PATROL17

1. SIGHTING0613
2. SIGHTING0594

QUESTION ? WHOSE SIGHTING is SIGHTING0613

1. RED

QUESTION ? WHOSE SIGHTING is SIGHTING0594

1. MERCHANT3

QUESTION ?
Leaving EXPLAIN.
```


066-18N
064-97N
063-69N
062-36N
060-96N
030-70W
027-62W
024-54W
021-46W
018-38W
RED
PATROL17
CONTACT1
CONTACT3
CONTACT2
BLUE

```
                *       *       *

  ***MESSAGE RECEIVED.  Message number 14

Time:  748
New message concerning PATROL17
Content:

  (SOURCE: EXTERNAL TOS: 645 LATITUDE: 61.77 LONGITUDE: -37.97)
Beyond area of interest.  Ignored.

        ;the patrol has now moved out of range.

                *       *       *


        ;at the end of the run, we summarize the conclusions.

Explanation system
Type HELP for help, CR to exit.
QUESTION ? WHAT is CONTACT3

1.  almost certainly not MERCHANT

QUESTION ? WHAT is CONTACT2

1.  definitely not MERCHANT

QUESTION ? WHAT is CONTACT1

1.  somewhat likely to be MERCHANT

QUESTION ?
Leaving EXPLAIN.
No more messages.
>QUIT
LEAVING STAMMERTOP
NIL
```


067-03N
065-42N
063-69N
061-86N
059-90N
037-19W
032-91W
028-64W
024-37W
020-09W
RED
PATROL17
PERRY
CONTACT1
CONTACT3
CONTACT2
BLUE


4-35N
3-44N
2-50N
1-52N
0-52N
031-08W
028-94W
026-81W
024-67W
022-53W
RED
PATROL17
PERRY
CONTACT1
MERCHANT6
CONTACT3
CONTACT2
BLUE

IV. USER'S GUIDE

DEMONSTRATION SYSTEM

To run the demonstration system, type RUN STAMMER[CR]. Answer the questions about terminal configuration and a map with Y for yes and N for no.

When a display appears, you may use any of the DSPLA function key commands listed in appendix A to manipulate the picture.

When the system enters explanation mode, you may ask questions to explore the data base and to trace the derivation of conclusions.

EXPLAIN is provided to allow you to ask questions about the contents of the data base (memory) and how they got there. Question forms are:

1. WHO is
2. WHAT is
3. WHOSE
4. WHERE is
5. WHY
6. Is
7. Tell me about
8. Display command level
9. OK
10. Zap
11. Help
12. New rule definition

Typing a carriage return to a QUESTION ? prompt will cause an exit from EXPLAIN.

Information in the data base is represented in the form

[ITEM] [RELATION] [ITEM]

Most explanation facilities are geared to letting you find out what is in one of these slots, given fillers of the others. For instance, WHO and WHAT return the first ITEM in all assertions whose RELATION and second ITEM are those given. WHOSE returns a second ITEM, given a RELATION and a first ITEM. Is will respond to the presence or absence of a fully specified assertion in the data base.

EXPLAIN has been designed to make user interface relatively easy. At most points, typing ? will give a list of possible next entries, and ESC may be used to complete unambiguous words, or fill in unambiguous characters. To help the system function properly, please give it time to do its job. Unacceptable characters are not echoed (the bell rings), as is the

case with excessive typeahead. If asked to [confirm], type a space. Please end questions with a carriage return.

WHO is

Format: WHO is (THE, A, AN) [RELATION] (OF) [ITEM]
[a, an, the, and of are optional]

Examples: WHO is LESS-THAN 7
WHO is INSIDE STORM0005

Some relations do not require that an item be specified. At present, these are INSIDE-A-MERCHANTLANE and REACHABLE-BY-A-COMBATANT. For these relations, the command looks like

WHO is INSIDE-A-MERCHANTLANE

For related queries, see WHOSE and WHAT.

WHAT is

Format: WHAT is (THE, AN, A) [RELATION] (OF) [ITEM]

Example: WHAT is THE LATITUDE OF POSITION0312

As with WHO, the, an, a, and of are optional, and an item is not required with the relations REACHABLE-BY-A-COMBATANT and INSIDE-A-MERCHANTLANE.

WHOSE

Format: WHOSE [RELATION] is [ITEM]

WHOSE is roughly the inverse of WHAT, e.g., if WHAT is THE POSITION OF SIGHTING0027 is answered with POSITION0026, then WHOSE POSITION is POSITION0026 will answer SIGHTING0027.

WHERE is

Format: WHERE is [OBJECT]

Example: WHERE is CONTACT7

Acceptable objects are platforms, merchantlanes, and storms.

WHY

Format: WHY [NUMBER]

Example: WHY 3

In reply to WHO, WHAT, WHOSE, and some Tell-me-about questions, you will be presented with a numbered list of answers. To follow the derivation of any of these, ask WHY followed by the number of the answer of interest. You will then be given a list of productions, if the answer was deduced by the system (not taken from a message or the technical data base). If such a list appears, you can do a WHY to it to view the information which enabled

the rule. The chain of WHYs may be extended indefinitely, alternating between data and rules. See the OK and Zap commands for further refinements.

Is

Formats: Is (THE, A, AN) [RELATION] (OF) [ITEM] [ITEM]
Is (THE) [ITEM] (A, AN, THE) [RELATION] (OF) [ITEM]
Is (THE) [PLATFORM] (A) [ID INFO]

Examples: Is THE LATITUDE OF POSITION0035 -1.22
Is RADAR THE SOURCE OF SIGHTING0342
Is KYNDA2 HOSTILE

Is checks to see if a given assertion is in the data base. A, an, the, and of are optional. Is is fairly flexible in format to allow more natural phrasing. In addition, identification information (name, class, category, flag, medium, type, hostility, etc.) is directly available, without the need to give the intervening relation.

Tell me about

Format: Tell me about [SOMETHING]
Tell me about [GROUP] [NUMBER]

Examples: Tell me about MERCHANTS
Tell me about RULE 5

Tell me about is probably the most flexible command. In the first format, you may ask about a wide range of things, including ITEMs, PRODUCTIONS (RULES), or any subtype (MERCHANTS, COMBATANTS, CONTACTS, PATROLS). Using the second format, you may examine the details of a particular message or rule (production) referred to by number.

Display command level

Format and example: Display command level

If you have a display available, this pseudo-query will give access to the top level of DSPLA to permit drawing rhumb lines and such like. If you are running on a Tektronix in single terminal mode, you will have to give the top level DSPLA command Q to return to EXPLAIN.

OK

Format and example: OK

OK is a useful part of the WHY feature. WHY functions by maintaining a context stack of "active" answers. OK pops this stack to allow you to ask about a different answer in a list you've already asked WHY of.

Zap

Format and example: Zap

Zap is the ultimate OK. It clears the WHY context stack completely.

HELP

Format and example: HELP

HELP calls the help function that prints a brief summary of each command, along with examples.

New rule definition

Format: New rule definition

Example: New rule definition

Rule name? RULE5

Text?

: IF *X IS A STORM

: AND *Y IS A CONTACTS

: AND *Z IS A SIGHTING OF *Y

: AND *Z IS INSIDE *X

: THEN MERCHANT IS THE CATEGORY OF *Y

Confidence? .35

RULE5

The name of a rule may be any LISP atom. Care should be taken to avoid conflicts with existing rule names, which may be listed using the Tell me about RULES query.

The text of a rule is made up of a number of lines, each of which is a condition or an action. No rule may have more than one action, and conditions must precede actions. Conditions are flagged with the words IF, AND, or UNLESS, and actions are flagged with the word THEN. Lines of a rule are generally of the form:

[flag word] [item] (IS, A, AN, THE) [relation] (OF) [item]

Items may be either variables or basenodes. Variables are flagged by beginning them with an asterisk. There are two other formats which may occur in rule lines. These are:

[flag word] [item] (IS, A, AN, THE) [special item] and

[flag word] [item] (IS) [special relation]

Special items are all members of SUBTYPES (CONTACTS, COMBATANTS, etc.) plus STORM and PLATFORM. Special relations are INSIDE-A-MERCHANTLANE and REACHABLE-BY-A-COMBATANT.

The confidence of a rule should be a number between 1.0 and −1.0. Use of recognition is supported during text input, and ? will respond with a list of expected entries, except during variable names.

To leave the system at an intermediate point, type two control-Cs. If you come to the end of the messages, type QUIT[CR] followed by control-C to exit.

Running From Scratch

The following files hold the code for the STAMMER system:

NEWSNET.COM

STAMMER.COM

EXPLAIN.COM

SUPP.COM

DSPLA.COM

FORK.COM

WITHINR.COM

These files must all be loaded (in the order given above) for STAMMER to function properly. *The easiest way to do this is to load the file FASTSAPS, which will load the other* files. This will give you an empty STAMMER, ready for initialization and experimentation.

The memory structure, contents, productions and messages which make up the demonstration system are contained in the files

STRUCT.COM

STMEM.LSP

STPROD.LSP, and

STMSG.LSP

respectively, and may be loaded directly. If you wish to load STAMMER with the demonstration scenario, load QKTEST in place of FASTSAPS.

Once the system is loaded, execute the function STAMMERTOP with no arguments to reach the STAMMER executive. Typing EXECUTE[CR] at this point will begin execution of the demonstration scenario.

V. DESIGN

The critical aspects of STAMMER's design may be divided into four parts. These are memory (the database), rule interpretation, explanation, and graphics. Following is a discussion of the design of each of these parts.

MEMORY

Memory in STAMMER consists of ASSERTIONS which are made up of two BASENODES and a RELATION. Assertions are of the form aRb (read "a is an R of b"), where a and b are the basenodes and R is a relation. The relations that may be used in assertions are restricted to a predefined (and expandable) set. It is possible to add assertions to and delete assertions from memory and to retrieve assertions and basenodes from memory.

The actual implementation of assertion structures is accomplished through a package of network manipulation functions called SNET and is relatively unimportant, with two exceptions. The first is that SNET creates a unique identifier for each assertion. Second, SNET labels each element of an assertion in the following way: for the example aRb, a is said to be ONE of the assertion, b the TWO, and R the RELATION. This terminology will be used hereafter.

The data base is divided into two sections, dynamic and virtual. Whether a particular assertion is dynamic or virtual depends on its relation. To save space, some common relations (such as LESS-THAN) are implemented as LISP functions, which take basenodes as arguments. These relations are called ORACLES, and are computed rather than retrieved from memory. The savings in space comes from eliminating the large number of assertions which would be required to represent, for example, all possible LESS-THAN relations. Assertions which have an oracle as their relation are said to be virtual, while those whose relations are not oracles are dynamic.

Every assertion has some confidence associated with it. This confidence is based on the source of the assertion. Assertions created by message receipt and oracles have confidences fixed at 1.0, while assertions created by rule firing have confidences based on their derivations. It is important to note a distinction between assertions and confidences. Assertions are statements about objects, while confidences are statements about assertions. For this reason, assertions may be referred to as first-order knowledge, and confidences as second-order knowledge.

There are a number of functions provided for manipulating assertions. These fall into the categories suggested by the operations of creation, deletion, and retrieval. Creation functions include ASSERT, CASSERT, STATE, and DENY. ERASE1 is the deletion function. Retrieval is performed by RETRIEVE2, RETRIEVE2B, RETRIEVE2S, RETRIEVE2BS, RETRIEVE3, and RETRIEVE3B.

The assertion functions have the property of never creating duplicates. If an assertion already exists, its identifier is returned as the result of a creation attempt.

RULE INTERPRETATION

Conceptually, the inference rules are very simple. Every rule attempts to retrieve information from memory and, if it succeeds, constructs a new assertion in memory. The retrievals are called conditions, and the construction specification is called the action. This view of rules is straightforward, but does not consider the complications introduced by variables and conditions that are not simple conjunctions.

Conditions may bind variables to the results of their retrievals. These bindings may be used by later conditions to constrain their retrievals. If retrievals had only one possible answer, variable binding would present no problem. However, many retrievals will return a number of answers from memory, and it is desirable to carry all of these possibilities forward so that one application of a rule will find all possible conclusions.

The method chosen for handling the problem of multiple bindings is backtracking. At every binding point (occurrence of a new variable), the possible values for the variable are collected (retrieved). One value is selected, and the remaining conditions are evaluated using a binding of the variable to that value. Upon failure of a condition or successful construction of a conclusion, evaluation backs up to the last binding point and selects a new value, then proceeds forward. When the possible bindings for a variable are exhausted, evaluation backs up to the preceding binding point. When the topmost binding point has finished with all possible values, evaluation of the rule is complete. Earlier, it appeared that a backtracking mechanism would be unnecessarily cumbersome, but clever application of recursive function calls and mapping functions have simplified the mechanism considerably.

There are several other aspects of rule evaluation that are important to the functioning of the system. The first is the interaction of confidence measures and the success of conditions. Conditions in STAMMER rules may be either positive or negative. Positive conditions succeed if there is any answer to a retrieval request with confidence greater than 0.0. Negative conditions fail if there is any answer to a retrieval request with confidence greater than 0.0. Failure to retrieve an assertion is assumed to be confidence 0.0. It is possible to remove the positive confidence restriction on positive conditions and still expect the system to work. However, doing so amounts to declaring the converses of all rules to be true, which unfortunately is not the case in general. Rather than accept cheap but occasionally faulty converses, the user is required to explicitly state converse rules.

Another issue relates duplication of conclusions and the construction of derivation trees. When a conclusion is added to the network (memory), a derivation tree containing the rule name and the assertions that satisfied the rule's conditions is added to the network and associated with the assertion. This derivation tree performs three functions. It is used to provide a trace of inferences for explanation; it is used to derive the confidence in the conclusion; and it serves to prevent duplications of conclusions. Before a conclusion is actually constructed, the would-be contents of its derivation tree are compared with the derivations of that assertion which already exist, if any. If any of the existing derivations matches the derivation under consideration, the conclusion is not constructed, and the existing binding set fails. This arrangement permits an assertion to be the conclusion of

several rules, or even of a single rule with different assertions satisfying its conditions, but prevents a rule from firing again and again based on the same evidence.

In the interest of efficiency, two further refinements have been added to the rule evaluation procedure. The first of these concerns rule selection, and the second limits the assertions that are considered by the conditions.

The original rule selection mechanism in STAMMER was a simple list of rule names, which were tested in order until a rule succeeded or all rules failed. When a rule succeeded, the process started again from the beginning of the list. If all failed, evaluation stopped. This meant that every time a rule fired, the rules at the beginning of the list were evaluated again. The first attempt to improve upon this inefficient process reorganized the rules into an effectively circular list, and instead of starting evaluation at the head of the list, upon achieving success merely went on to the next rule in order. Evaluation was halted when all rules had been evaluated without succeeding since the last success. While this change brought some improvement, the increased speed was primarily an artifact of the particular rule set in use, since the two arrangements are essentially identical, being linear searches of a list. A major improvement has been made by adding information about the rules to the evaluation process. Certain rules, once tried, will never again be satisfied until another message or report is received. After evaluating these rules once, they can be (temporarily) removed from the rule list, thus decreasing the length of the list that is tested sequentially, resulting in a noticeable improvement in execution time. However, this improvement is possible only through the inclusion of "meta" information about the structure of rules and particular rule sets. This information is not derived automatically at present, but must be added by hand to appropriate rules.

Another type of meta information is used automatically to reduce the number of assertions that are considered by each condition. The idea underlying this refinement (which has resulted in a better than ten-to-one increase in execution speed) is one of context. At any point in the system's execution, many assertions need not be considered as candidates for conditions because they have been used in the past. However, distinguishing "interesting" from "uninteresting" assertions is no easy task. The following steps illustrate the reasoning behind the context mechanism that STAMMER employs:

1) Consider a state of the network. Try to fire rules. Some may succeed, others fail. Keep trying until all fail. At this point, no rule will fire until new assertions are added to memory.

2) At the start of a STAMMER run, no rules can fire until a message or report is received. (This corresponds roughly to the base step of an induction, with the rest of the points providing the induction step.)

3) Add some new assertions through message receipt. Now the only rule firings that can occur are those that use the new assertions to satisfy at least one of their conditions.

4) For those conditions satisfied by new assertions, old assertions may also provide possible bindings. But if the bindings provided by old assertions were to succeed, they would have done so before the new assertions arrived. They didn't so you can ignore the bindings provided by old assertions in conditions where new assertions are available.

With some minor adjustments, this argument summarizes the approach used. The message monitor and the conclusion building mechanism add their assertions to the context, which is cleared before each message receipt. In evaluating a condition, its retrieval is performed. If any new assertion is included in the retrieval, only new assertions are considered for that condition. Otherwise, all retrievals are considered.

In practical terms, this means that the system concentrates on what it just learned about, without wasting time trying possibilities that failed in the past and haven't changed. The improvement in performance is significant.

This method of controlling ineffective inferences works only so long as all rules have a chance to fire at all times (on all new information receipts). If a new rule is defined in the middle of a run, it would not have a chance to be applied to information that was added to memory before its definition. To avoid missing inferences when new rules are defined, part of the rule-definition procedure is an application of the new rules to the existing memory with the context cleared. Since no assertions are marked as new, no "new" assertions can block the use of old assertions in satisfying the rule, so all possible conclusions are derived.

EXPLANATION

The explanation system provides two primary capabilities – retrieval of memory contents and inference tracing. In support of the user interface, there are two features of the system that have been concentrated on in design.

Retrieving memory contents is a relatively easy task, since functions for the retrievals already exist. Similarly, since derivations are also a part of the system memory (created by conclusions), tracing a derivation presents no major computational problems. The problem in explanation lies in providing a "comfortable" format for the user to phrase queries. To make the user interface as natural as possible, the explanation system provides a query language that is "English-like." This language is an extremely limited version of English, which includes only certain types of questions and methods of phrasing those questions. However, the language was designed so that, while limited, it is sufficient to cover the user's needs without making its shortcomings apparent.

Using a natural language-like language is helpful in encouraging an intuitively simple user interface, but obviously creates some problems. The problem of parsing can be dealt with by careful language design. Another problem, which is more "human factors" oriented, is the large amount of typing which an English-like language can require. To avoid this problem, the explanation system makes heavy use of the LISP function ASKUSER, which features recognition, prompting, and other features conducive to pleasant interaction. The combination of a carefully designed language and ASKUSER makes it possible to indicate the form of retrievals simply.

The second feature of the user interface which simplifies interaction is the use of a "prettyprinter" for memory contents. Rather than answering retrieval requests with a (probably) incomprehensible network structure, all answers are printed in a format designed to be easily understood. Assertions are printed with noise words like "is" and "of", and confidences are converted into modifier strings. When a list of answers is presented, the list is numbered, with each element presented on a separate line. Basenodes are printed in a form that is conceptually clearer than their internal representation (e.g., SIGHTING0732 rather than N0732).

Perhaps the most important aspect of the explanation system is that, on request, help is available to explain the use of the explanation system itself.

GRAPHICS

Graphical support for STAMMER is provided by DSPLA (Ref. 1), a package developed at NOSC specifically for tactical situation assessment. The DSPLA system is a collection of FORTRAN subroutines that allow storage, retrieval, and display of ship and aircraft tracks. It provides many tools for manipulating and enhancing the usefulness of the displays. STAMMER controls the display routines by means of a LISP-FORTRAN interface, obtained from SRI International and modified for our purpose. This makes use of TOPS-20 monitor calls to establish the display routines as a separate inferior process to the dominant INTERLISP process running STAMMER. The interface is facilitated by a set of LISP procedures tailored especially for controlling DSPLA. These include additional monitor calls that allow the use of a separate terminal for graphics output.

Modifications to DSPLA

Integrating DSPLA with STAMMER required some alterations to the code and link-editing procedure. These were necessary (1) to support the LISP-FORTRAN interface, (2) to attach a new front end to STAMMER, and (3) to make minor improvements to the basic system. In general these changes are not visible at the user command level.

To support the interface, the top level control loop was changed to a single call and return to allow control to pass back to STAMMER. Since the interface does not permit calling a main FORTRAN program, all essential control was transferred to subroutines. For convenience some input normally derived from user interaction is now instead passed as arguments to subroutines (here the STAMMER controlling process takes the place of the user). At link-edit time, it is necessary to include a support file F10F.REL and perform an initialization procedure (Details are in the file FORK.DOC. See below.)

Attaching the front end to STAMMER required subroutines to initialize an empty data base and to add new platforms and sightings as they arose. Because of limitations in space available and its disuse in our system, the ground truth structure was shrunk to nothing. Simultaneously the tracking structure was expanded to accommodate the large number of tracks required by our system. To allow saving the DSPLA structure for later restart, the initialization had to be non-destructive, i.e., accomplished by means of data statements.

Other minor changes to DSPLA include new input routines to explicitly set the correct character echoing mode for TOPS-20 and to obviate the need for a carriage return when in finger key mode. The algorithm for computing the window that includes a given set of tracks was modified to preserve the aspect ratio. This was to maintain compatibility with the ISI map routine which otherwise arbitrarily contracts the picture. The routines that search the tracking structure for platforms were modified where necessary to require a match on all ten characters. This change allowed the use of platform names which coincide in the first five characters. To allow graphic output to go to a terminal other than the job controlling one, a new global variable and common storage area, NTTY, was introduced.

Further information on these changes is given in comments in the code.

Modifications to LISP-FORTRAN Interface

Modifications were necessary to allow Fortran-10 (in which DSPLA is written) to be used in place of Fortran-4 (also known as F40). The interface was also changed so that the superior and inferior processes could run in parallel, with synchronizing waits only where necessary. This feature is very useful when running with graphics going to a separate display terminal.

GRAPHICS-RELATED FILES

LISP-FORTRAN interface: FORK.LISP, FIOF.MAC (see also FORK.DOC).

DISPLA routines: DISPLIB. FOR, DISPIO.MAC

STAMMER-DISPLA Interface: DISPLA.LSP

VI. MATHEMATICAL TECHNIQUES

Mathematical methods are used extensively only in the geometry procedures that determine the geographical conditions satisfied by sightings. The functions that combine confidences also merit some discussion.

GEOMETRY FUNCTIONS

Distance and bearing are computed by formulas derived from three-dimensional vector geometry. Thus they are not flat-Earth approximations, but are valid for all points of the Earth's surface. Higher level functions use well-known trigonometric formulas, based on a flat-Earth approximation. However, they are defined in terms of distance and bearing, so that their results are applicable to all regions of the Earth sufficiently small to be treated as flat, including regions near the poles.

A couple of procedures are deserving of special mention. The function that determines interior points of a polygon is based on Cauchy's Integral Formula. Functions that determine whether a platform could have reached a given position without being spotted by a patrol use geometric methods rather than algebraic or analytic ones.

DERIVATIONS OF GEOMETRIC FORMULAS

1. Range, bearing

Assume the Earth is a sphere. With suitable units, the radius may be taken as 1. The spherical polar coordinates of a point on the surface are $(1, \theta, \phi)$, where θ and ϕ are the latitude and longitude, respectively. In Cartesian coordinates this becomes

$$(\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta).$$

Let A and B be points with lat-lon pairs (θ_1,ϕ_1) and (θ_2,ϕ_2) respectively. Let u and v be the position vectors of A and B. Let ψ be the angle between u and v, that is, the angle subtended at the Earth's center by A and B. Then

$$\begin{aligned}\cos\psi &= u\cdot v\\ &= \cos\theta_1\cos\phi_1\cos\theta_2\cos\phi_2+\cos\theta_1\sin\phi_1\cos\theta_2\sin\phi_2+\sin\theta_1\sin\theta_2\\ &= \cos\theta_1\cos\theta_2(\cos\phi_1\cos\phi_2+\sin\phi_1\sin\phi_2) + \sin\theta_1\sin\theta_2\\ &= \cos\theta_1\cos\theta_2\cos(\phi_1-\phi_2) + \sin\theta_1\sin\theta_2\end{aligned}$$

The range in nautical miles is given by $60*\psi$ (when ψ is measured in degrees).

To compute the bearing from A to B, we first obtain a unit vector p, orthogonal to both u and v. This is given by

$$p = (u \times v)/\sin\psi.$$

Next we require a unit vector q orthogonal to u, but in the plane determined by u and the

Earth's axis. We may write

$$q = [(u \times k)/\cos\theta_1] \times u,$$

where k is a unit vector in the z direction (parallel to the Earth's axis, pointing north).

The vector q may be positioned tangent to the Earth at A, pointing northward. The vector p can be placed so that it is tangent to the Earth at A and rotated 90 deg counterclockwise from the direction toward B. Then (modulo 360)

$$\text{Bearing}(A,B) = 90 \pm \arccos(p \cdot q).$$

The ambiguous sign is positive when the rotation from p to q is counterclockwise.

We derive an expression for $p \cdot q$ as follows:

$$p \cdot q = [(u \times k) \times u] \cdot (u \times v)/(\cos\theta_1 \sin\psi)$$

$$\begin{aligned}
\text{But } [(u \times k) \times u] \cdot (u \times v) &= -[(k \times u) \times u] \cdot (u \times v) \\
&= -k \cdot (u \times [u \times (u \times v)]) \\
&= -k \cdot (u \times v), \text{ since u is a unit vector} \\
&= -(\cos\theta_1 \cos\phi_1 \cos\theta_2 \sin\phi_2 - \cos\theta_1 \sin\phi_1 \cos\theta_2 \cos\phi_2) \\
&= \cos\theta_1 \cos\theta_2 \sin(\phi_1 - \phi_2)
\end{aligned}$$

If we assume $\theta_1 \neq \pm 90$, we may cancel $\cos\theta_1$, to get

$$p \cdot q = \cos\theta_2 \sin(\phi_1 - \phi_2)/\sin\psi$$

The bearing is undefined when $\psi = 180$. We set

$$\text{Bearing}(A,B) = \begin{cases} 90 \text{ when } \theta_1 = 90 \\ 0 \text{ when } \theta_1 = -90 \end{cases}$$

2. Interior points of a polygon

If Γ is a simple closed rectifiable curve and a is any complex constant, then

$$\int_\Gamma \frac{dz}{z-a} = \begin{cases} 2\pi i, \text{ if a is inside } \Gamma \\ 0 \text{ otherwise} \end{cases}$$

by Cauchy's Integral Formula (Ref. 3).

$$\begin{aligned}
\text{But } \int_\Gamma \frac{dz}{z-a} &= \int_\Gamma d(\log(z-a)) \\
&= \int_\Gamma d(\log|z-a|) + i\int_\Gamma d(\arg(z-a))
\end{aligned}$$

When Γ is completely traversed, $\log|z-a|$ returns to its original value.

Thus $\int_\Gamma d(\log|z-a|) = 0$

It follows that (cancelling the i)

$$\int_{\Gamma} d(\arg(z-a)) = \begin{cases} 2\pi, & \text{if } a \text{ is inside } \Gamma \\ 0 & \text{otherwise} \end{cases}$$

Suppose Γ is a polygon with vertices (in order) $z_1, \ldots, z_n$. For convenience we write $z_{n+1} = z_1$. Then

$$\int_{\Gamma} d[\arg(z-a)] = \sum_{j=1}^{n} \int_{z_j}^{z_{j+1}} d[\arg(z-a)]$$

$$= \sum_{j=1}^{n} [\arg(z_{j+1}-a) - \arg(z_j-a)],$$

where for each summand, a branch of arg is chosen that is continuous in the region of integration. If we write

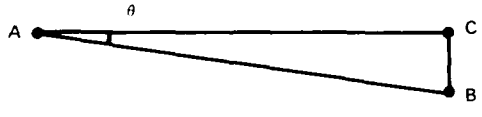
b_j = Bearing(a,z_{j+1}) - Bearing(a,z_j), then

$$\arg(z_{j+1}-a) - \arg(z_j-a) = \begin{cases} 360+b_j & \text{if } b_j < -180 \\ b_j & \text{if } -180 \leq b_j \leq 180 \\ b_j-360 & \text{if } b_j > 180 \end{cases}$$

3. Closest approach

Suppose S_1 and S_2 are objects moving at constant velocity. Assume S_1 starts from A at time 0 with velocity vector v_1. Simultaneously S_2 starts from B with velocity vector v_2. We wish to compute d, the distance of closest approach and t, the time of closest approach.

Let C be the point of closest approach. Relative to S_2, the motion appears as follows:



Relative to S_2, it appears that S_1 moves with velocity v_1-v_2. Let w be a unit vector in the direction from A to B. Then

$$\cos\theta = w\cdot(v_1-v_2)/|v_1-v_2|$$

Clearly $d = \text{Range}(A,B)\sin\theta$

and $t = \text{Range}(A,B)\cos\theta/|v_1-v_2|$

CONFIDENCE COMBINING FUNCTIONS

These are the same as those used in MYCIN (Ref. 2) except that disjunction does not occur in our rules. Briefly, each assertion has a measure of belief and a measure of disbelief associated with it. The antecedent of each rule is a conjunction of conditions. The measure of belief (disbelief) of the antecedent is the minimum (maximum) of the

measures of belief (disbelief) of the individual conditions. Each rule has a weight attached to it. The contribution of belief (disbelief) provided by a successful rule firing is the product of this weight and the measure of belief (disbelief) of the antecedent. When two rules bear upon the same hypothesis, their contributions of belief and disbelief are combined separately according to the formula

$$\text{Combine}(c_1,c_2) = c_1 + c_2 - c_1 * c_2.$$

Then the total contribution of disbelief is subtracted from the total contribution of belief to give the confidence. When more than two rules combine, the process is iterated. That is,

$$\text{Combine}(c_1,c_2,c_3) = \text{Combine}[c_1,\text{Combine}(c_2,c_3)].$$

REFERENCES

1. Naval Ocean Systems Center, "DSPLA – A Graphics Package for Tactical Situation Assessment. (Version II)," by G. G. McIntyre, NOSC Technical Note 530, September 15, 1978.
2. Computer-Based Medical Consultations: MYCIN. E. H. Shortliffe. American Elsevier, New York, 1976.
3. E. Hille, *Analytic Function Theory*, vol. 1, Blaisdell, New York, 1959.

Appendix A

DSPLA Commands

FUNCTION KEYS ARE ENTERED AS ONE LETTER. TO OBTAIN INFORMATION ON SPECIFIC FUNCTION KEYS, RETURN TO COMMAND MODE AND EXECUTE <FKDOC>.

FK	DESCRIPTION
P	POSITION TO STACK
M	MAGNIFY ABOUT CURSOR
R	REDUCE ABOUT CURSOR
V	VIEW ABOUT CURSOR
L	LIST FILES AVAILABLE
G	LIST GROUND TRUTH
T	LIST TRACKING
H	LIST HISTORY
D	RE-DISPLAY PLOT
8	SET TYPE SIZE 1
9	SET TYPE SIZE 2
:	SET TYPE SIZE 3
;	SET TYPE SIZE 4
?	HELP
C	RETURN TO COMMAND MODE
K	POSITION TO TRACK STACK
N	MOVE LAST FIGURE FWD
	REPEAT LAST PLOT COMND
S	SET GLOBAL COORDINATES
A	APPEND TO FOLLOWING
Z	CLEAR TRACK STACK
F	MOVE TIME FORWARD
B	MOVE TIME BACKWARD
U	LIST FIGURE FILE
E	EXIT FK MODE

'P' FUNCTION KEY

POSITION THE CURSOR ANYWHERE ON THE TERMINAL SCREEN AND ENTER 'P'. THE LAT-LON POSITION OF THE CURSOR WILL BE PLACED INTO A STACK WHICH CAN HOLD 10 POSITIONS (THE 11TH POSITION WILL OVER-WRITE THE 1ST). THE POSITIONS IN THE STACK CAN THEN BE USED FOR OTHER PURPOSES, SUCH AS POINTS OF ORIGIN FOR BEARING LINES, CENTERS OF FIGURES, ETC.

'M' FUNCTION KEY

POSITION THE CURSOR ANYWHERE ON THE TERMINAL SCREEN AND ENTER 'M'. THE DISPLAY WILL BE MAGNIFIED BY A FACTOR SET BY <SCALE> AND CENTERED AT THE CURSOR POSITION. THE DEFAULT FACTOR IS 2.

'R' FUNCTION KEY

POSITION THE CURSOR ANYWHERE ON THE TERMINAL SCREEN AND ENTER 'R'. THE DISPLAY WILL BE REDUCED BY A FACTOR SET BY <SCALE> AND CENTERED AT THE CURSOR POSITION. THE DEFAULT FACTOR IS 2.

'V' FUNCTION KEY

POSITION THE CURSOR ANYWHERE ON THE TERMINAL SCREEN AND ENTER 'V'. THE DISPLAY WILL BE GIVEN AT THE SAME SCALE BUT CENTERED AT THE CURSOR POSITION.

'L' FUNCTION KEY

ENTER 'L' AND A LISTING OF ALL ACCESSABLE FILES AND SUBFILES WILL BE GIVEN ALONG WITH THEIR CORRESPONDING FILE CODES. WHEN PROMPTED, ENTER ONE OF THE FILE CODES (SUBFILES MAY BE VIEWED ONLY WITHIN ITS PARENT FILE). THE FILE WILL BE DISPLAYED WITH PAGE SIZE SET BY <PGSIZ> (DEFAULT IS 20 LINES) AND CURRENT TYPE SIZE (MUST BE SIZE 1 FOR SINGLE TYPE SIZE TERMINALS).

'G' FUNCTION KEY

ENTER 'G' TO VIEW THE GTH FILE. PAGE SIZE IS AS SET BY <PGSIZ> AND CURRENT TYPE SIZE IS USED (MUST BE SIZE 1 FOR SINGLE TYPE SIZE TERMINALS). ONCE THE FILE IS DISPLAYED, USE 'F' TO VIEW SUCCEEDING PAGES; 'B' TO VIEW PRECEEDING PAGES; 'R' TO RETURN FROM VIEWING, 'E' TO EDIT A LINE; AND ANY VALID SUBFILE CODE TO VIEW SUBFILES.

'T' FUNCTION KEY

ENTER 'T' TO VIEW THE TRH FILE. PAGE SIZE IS AS SET BY <PGSIZ> AND CURRENT TYPE SIZE IS USED (MUST BE SIZE 1 FOR SINGLE TYPE SIZE TERMINALS). ONCE THE FILE IS DISPLAYED, USE 'F' TO VIEW SUCCEEDING PAGES; 'B' TO VIEW PRECEEDING PAGES; 'R' TO RETURN FROM VIEWING, 'E' TO EDIT A LINE; AND ANY VALID SUBFILE CODE TO VIEW SUBFILES.

'H' FUNCTION KEY

POSITION THE CURSOR AT THE LOWER LEFT CORNER OF THE FIRST LETTER OF THE TRACK NAME AND ENTER 'H'. THE TRACK HISTORY OF THAT TRACK WILL BE DISPLAYED IN FILE FORMAT. CONTINUE WITH FILE COMMANDS.

'D' FUNCTION KEY

IF IN FUNCTION KEY MODE THE DISPLAY IS ERASED FOR VIEWING OTHER DATA, ENTER 'D' TO RE-DISPLAY THE PLOT.

'8' FUNCTION KEY

ENTER '8' TO SET TYPE SIZE TO LARGEST ON THE TEKTRONIX 4014 TERMINAL. THIS IS THE DEFAULT TYPE SIZE AND SHOULD BE USED ON SINGLE TYPE SIZE TERMINALS.

'9' FUNCTION KEY

ENTER '9' TO SET TYPE SIZE TO NEXT TO LARGEST ON THE TEKTRONIX 4014 TERMINAL. THIS SIZE SHOULD NOT BE USED ON SINGLE TYPE SIZE TERMINALS.

':' FUNCTION KEY

ENTER ':' TO SET TERMINAL TYPE SIZE TO NEXT TO SMALLEST ON THE TEKTRONIX 4014 TERMINAL. THIS SIZE SHOULD NOT BE USED ON SINGLE TYPE SIZE TERMINALS.

';' FUNCTION KEY

ENTER ';' TO SET TYPE SIZE TO SMALLEST ON THE TEKTRONIX 4014 TERMINAL. THIS SIZE SHOULD NOT BE USED ON SINGLE TYPE SIZE TERMINALS.

'?' FUNCTION KEY

ENTER '?' TO OBTAIN INFORMATION ON ENTRY, FORMAT, AND HOW TO OBTAIN ADDITIONAL INFORMATION ON FUNCTION KEYS, AS WELL AS A LISTING OF AVAILABLE FUNCTION KEYS AND THEIR FUNCTIONS.

'K' FUNCTION KEY

POSITION THE CURSOR ANYWHERE ON THE TERMINAL SCREEN AND ENTER 'K'. THE LAT-LON POSITION IS PLACED IN A TRACK STACK WHERE IT MAY BE INTERPRETED AS A TRACK INCIDENT. INCIDENTS ARE SEPARATED BY A TIME INCREMENT, DT (SET BY <T><SP><I>). 'Z' WILL CLEAR THE TRACK STACK OF ALL DATA. <WRITE> MAY BE USED TO WRITE THE TRACK STACK ON TO THE DISK AS A FILE; <EXCH> MAY BE USED TO EXCHANGE THE TRACK STACK WITH A TRACK IN THE TRACKING STRUCTURE; OR <READ> MAY BE USED TO READ A FILE CONTAINING TRACK DATA INTO THE TRACK STACK. POINTS THEN PLACED INTO THE STACK BY 'K' WILL FOLLOW THE DATA ALREADY THERE.

'S' FUNCTION KEY

ENTER 'S' TO CAUSE THE GLOBAL AREA COORDINATES TO BE SET TO THE CURRENTLY DISPLAYED AREA WINDOW.

'A' FUNCTION KEY

ENTER 'A' TO APPEND CURRENTLY DISPLAYED TRACKS TO ALL FOLLOWING DISPLAYS. THIS CAN ONLY BE CANCELLED BY <NOAPP>. APPENDED TRACKS ARE UNAFFECTED BY CHANGES IN THE TIME WINDOW.

'Z' FUNCTION KEY

ENTER 'Z' TO CLEAR THE TRACK STACK OF ALL DATA.

'F' FUNCTION KEY

ENTER 'F' TO MOVE THE TIME WINDOW FORWARD BY A TIME INCREMENT, DT (SET BY <T><SP><I>) AND EXECUTE THE PREVIOUS PLOT COMMAND.

'B' FUCNTION KEY

ENTER 'B' TO MOVE THE TIME WINDOW BACKWARD BY A TIME INCREMENT, DT (SET BY <T>><SP><I>) AND EXECUTE THE PREVIOUS PLOT COMMAND.

' ' FUNCTION KEY (SPACE)

ENTER ' ' TO EXECUTE THE PREVIOUS PLOT COMMAND.

'N' FUNCTION KEY

ENTER 'N' TO CAUSE THE MOST RECENTLY DEFINED FIGURE (CIRCLE, ELLIPSE, ETC.) TO HAVE ITS POINT OF ORIGIN OR CENTER CHANGED TO THE TOP POSITION IN THE TRACK STACK.

'C' FUNCTION KEY

ENTER 'C' TO RETURN TO COMMAND MODE.

'U' FUNCTION KEY

ENTER 'U' TO VIEW THE FIGURE FILE. PAGE SIZE IS AS SET BY <PGSIZ> AND CURRENT TYPE SIZE IS USED (MUST BE SIZE 1 FOR SINGLE TYPE SIZE TERMINALS). ONCE THE FILE IS DISPLAYED, USE 'F' TO VIEW SUCCEEDING PAGES; 'B' TO VIEW PRECEEDING PAGES; 'R' TO RETURN FROM VIEWING; 'E' TO EDIT A LINE; AND 'Q' TO QUIT VIEWING.

'E' FUNCTION KEY

ENTER 'E' TO EXIT FUNCTION KEY MODE AND RETURN TO KEYBOARD MODE.

TYPE COMMANDS IN THE FOLLOWING FORM <COMMAND> <SPACE> <ARGUMENTS>. TO OBTAIN INFORMATION ON A SPECIFIC COMMAND AND ITS ARGUMENTS, TYPE <COMMAND> <SPACE> <?>.

COMMAND	DESCRIPTION
PGT	PLOT FROM GROUND TRUTH
Q	QUIT
LGT	LOCATE FROM GROUND TRUTH
G	SET GEOGRAPHIC MAP
M	MAGNIFY DISPLAY
F	MOVE TIME FORWARD
A	CHANGE AREA WINDOW
T	CHANGE TIME WINDOW
S	STORE DISPLAY
D	RECALL DISPLAY
K	KILL DISPLAY
E	ENQUIRE
R	REDUCE DISPLAY
	REPEAT LAST PLOT
C	SET GEOGRAPHIC AREA
APPND	APPEND TO FOLLOWING
NOAPP	CANCEL APPEND
START	RE-START DSPLA
CELL	DEFINE ELLIPSE
LOB	DEFINE BEARING LINE
CIRCLE	DEFINE CIRCLE
PTR	PLOT FROM TRACKING
LTR	LOCATE FROM TRACKING
FILES	LIST FILES AVAILABLE
SIZE3	SET TYPE SIZE 3
SIZE1	SET TYPE SIZE 1
SIZE2	SET TYPE SIZE 2
SIZE4	SET TYPE SIZE 4
PGSIZE	SET LINES PER PAGE
?	HELP
SCALE	SET MAGNIFY SCALE
ARROW	INCLUDE ARROWHEADS
NOARR	NO ARROWHEADS
LABEL	INCLUDE LABELS
NOLBL	NO LABELS
SYM	INCLUDE SYMBOLS
NOSYM	NO SYMBOLS
LINES	CONNECT WITH LINES

NOLIN	NO LINES
B	MOVE TIME BACKWARD
RHUMB	DEFINE RHUMB LINE
FKDOC	FUNCTION KEY INFO
RADIUS	DEFINE RADIUS OF TRAVEL
WRITE	WRITE TRACK STACK
READ	READ TRACK FROM DISK
EXCH	EXCHANGE TRACK STACK
HLPFL	ACCESS HELP FILE
RBT	FIND RNG, BNG, TIME
RFILE	READ TRH/TRK, GTH/GTK FILES
WFILE	WRITE TRH/TRK, GTH/GTK FILES
DISP	RE-DISPLAY PREVIOUS PLOT
GTH	DISPLAY GTH FILE
TRH	DISPLAY TRH FILE
FIG	DISPLAY FIGURE FILE
FKMODE	ENTER FUNCTION KEY MODE

FORM OF DISPLAY FIGURE FILE: <FIG>

THE FIGURE FILE IS DISPLAYED FOR VIEWING AND/OR EDITING USING COMMANDS OF THE FILE HANDLING SYSTEM. FOR A DESCRIPTION OF THESE COMMANDS, EXECUTE <FILES> AND THEN '?'.

FORM OF ENTER FUNCTION KEY MODE: <FKMOD>

THE ATTACHED PROGRAM FUNCTION KEYBOARD BECOMES THE PRIMARY COMMAND SOURCE. COMMANDS ARE EXECUTED BY POSITIONING THE CURSOR (WHEN NECESSARY) AND PUSHING THE APPROPRIATE FUNCTION KEY BUTTON. FUNCTION KEY MODE IS EXITED WHEN THE "KBMODE" FUNCTION KEY IS PUSHED.

FORM OF WRITE FILE: <WFILE><SP><FILE NAME>

FILE NAME IS A 5 CHARACTER (MAX) ALPHAMERIC NAME WHICH WILL IDENTIFY THE FILE TO BE WRITTEN. FILE NAMES BEGINNING WITH 'G' WILL CAUSE DATA TO BE WRITTEN FROM THE GTH/GTK STRUCTURES. ALL OTHER FILE NAMES DENOTE THE TRH/TRK STRUCTURES. THE FILE IS WRITTEN TO DISK IN BINARY MODE.

FORM OF READ FILE: <RFILE><SP><FILE NAME>

FILE NAME IS A 5 CHARACTER (MAX) ALPHAMERIC NAME WHICH WILL IDENTIFY THE FILE TO LE READ. FILE NAMES BEGINNING WITH 'G' WILL CAUSE DATA TO BE READ INTO THE GTH/GTK STRUCTURES. ALL OTHER FILE NAMES DENOTE THE TRH/TRK STRUCTURES. THE DISK FILE TO BE READ SHOULD HAVE BEEN CREATED BY DSPLA USING THE <WFILE> COMMAND.

FORM OF RE-DISPLAY PREVIOUS PLOT: <DISP>

THE PREVIOUS DISPLAY IS REPRODUCED FOR VIEWING AND/OR ACTION.

FORM OF DISPLAY GTH FILE: <GTH>

THE GTH FILE IS DISPLAYED FOR VIEWING AND/OR EDITING USING COMMANDS OF THE FILE HANDLING SYSTEM. FOR A DESCRIPTION OF THESE COMMANDS, EXECUTE <FILES> AND THEN '?'.

FORM OF DISPLAY TRH FILE: <TRH>

THE TRH FILE IS DISPLAYED FOR VIEWING AND/OR EDITING USING COMMANDS OF THE FILE HANDLING SYSTEM. FOR A DESCRIPTION OF THESE COMMANDS, EXECUTE <FILES> AND THEN '?'.

FORM OF FIND RNG, BNG, TIME: <RBT>

DATA IS DRAWN FROM THE TRACK STACK. WITH A GIVEN SPEED, THE RANGE, BEARING, AND TIME OF TRAVEL FROM POINT 1 TO POINT 2, POINT 2 TO POINT 3, ETC. IS CALCULATED, USING GREAT CIRCLE FORMULA, AND DISPLAYED.

FORM OF SET MAGNIFY SCALE: <SCALE>

ALLOWS CHANGING THE FACTOR BY WHICH DISPLAYS ARE MAGNIFIED OR REDUCED. THE DEFAULT FACTOR IS 2.

FORM OF INCLUDE ARROWHEADS: <ARROW>

CAUSES ARROWHEADS TO BE DRAWN AT THE ENDS OF TRACK LINE SEGMENTS TO DENOTE THE DIRECTION OF TRAVEL OF TRACKS WHEN PLOTTED (DEFAULT CONDITION).

FORM OF NO ARROWHEADS: <NOARR>

CAUSES ARROWHEADS TO NOT BE INCLUDED ON TRACKS WHEN PLOTTED.

FORM OF INCLUDE LABELS: <LABEL>

CAUSES THE TRACK NAME TO BE DISPLAYED BY THE TRACK WHEN PLOTTED (DEFAULT CONDITION).

FORM OF NO LABELS: <NOLBL>

CAUSES THE TRACK NAME TO NOT BE DISPLAYED BY THE TRACK WHEN PLOTTED.

FORM OF INCLUDE SYMBOLS: <SYM>

CAUSES SYMBOLS TO BE DRAWN AT TRACK INCIDENT POINTS WHEN TRACK IS PLOTTED (DEFAULT CONDITION).

FORM OF NO SYMBOLS: <NOSYM>

CAUSES SYMBOLS TO NOT BE DRAWN AT TRACK INCIDENT POINTS WHEN TRACK IS PLOTTED.

FORM OF NO LINES: <NOLIN>

CAUSES TRACK INCIDENT POINTS TO NOT BE CONNECTED WITH LINES WHEN A TRACK IS PLOTTED.

FORM OF CONNECT WITH LINES: <LINES>

CAUSES TRACK INCIDENT POINTS TO BE CONNECTED WITH LINES WHEN TRACK IS PLOTTED (DEFAULT CONDITION).

FORM OF MOVE TIME BACKWARD: <B>

MOVES THE TIME WINDOW BACKWARD BY A TIME INCREMENT DT (SET BY <T><SP><I>), AND THE PREVIOUS PLOT COMMAND IS EXECUTED (<SP>).

FORM OF DEFINE RHUMB LINE: <RHUMB>

ALLOWS DEFINING A LINE OF BEARING ALONG A RHUMB LINE AND SETTING A SYMBOLIC NAME FOR THAT BEARING LINE. PARAMETERS REQUIRED ARE POINT OF ORIGIN AND BEARING. THE SYMBOLIC NAME MAY THEN BE USED AS THE NAME OF A TRACK WHICH PLOTS THE BEARING LINE.

FORM OF FUNCTION KEY INFO: <FKDOC>

ALLOWS ACCESS TO DOCUMENTATION ON AVAILABLE FUNCTION KEYS.

FORM OF DEFINE RADIUS: <RADIU>

ALLOWS DEFINING A RADIUS OF TRAVEL AND SETTING A SYMBOLIC NAME FOR THAT RADIUS. PARAMETERS REQUIRED ARE POSITION OF CENTER AND SPEED. THE TIME INCREMENT, DT (SET BY <T><SP><I>), IS MULTIPLIED BY THE SPEED TO GIVE A RADIUS. THE SYMBOLIC NAME MAY THEN BE USED AS THE NAME OF A TRACK WHICH PLOTS THAT RADIUS.

FORM OF WRITE TRACK STACK: <WRITE><SP><FILE NAME>

FILE NAME IS UP TO 5 ALPHAMERIC CHARACTERS WHICH ARE TO IDENTIFY THE TRACK IN THE FORM OF A DISK FILE. THE EXTENSION OF FILE NAME WILL BE .DAT. DATA ASSOCIATED WITH THE TRACK IN THE TRACK STACK (NAME, TARGET PARAMETER, TYPE, NUMBER OF INCIDENTS, AND TRACK NUMBER) ARE ALSO WRITTEN. IF TRACK NAME IS UNDEFINED, A NAME, INITIAL TRACK TIME, AND TRACK NUMBER MUST BE ENTERED BEFORE THE TRACK IS WRITTEN. THE TRACK STACK IS UNAFFECTED BY THE WRITE OPERATION.

FORM OF READ TRACK FROM DISK: <READ><SP><FILE NAME>

FILE NAME IS THE 5 CHARACTER (MAX) ALPHAMERIC DISK FILE NAME WHICH IDENTIFIES THE TRACK TO BE READ. AFTER READING THE TRACK STACK WILL CONTAIN THE TRACK AND ASSOCIATED DATA (TRACK NAME, TYPE, ETC.). POINTS ENTERED INTO THE TRACK STACK WILL FOLLOW THE DATA HELD THERE.

FORM OF EXCHANGE TRACK STACK: <EXCH><SP><TRACK NAME>

TRACK NAME IDENTIFIES A TRACK ALREADY IN THE TRACKING STRUCTURE AND MAY BE THE NAME, TYPE OR CLASS, OR TARGET PARAMETER. THAT TRACK IS THEN EXCHANGED WITH THE TRACK IN THE TRACK STACK, ALONG WITH ALL ASSOCIATED DATA. PLOTTING MAY THEN BE DONE WITH THE NEW TRACK OR THE OLD TRACK NOW IN THE TRACK STACK MAY BE WRITTEN TO DISK. POINTS ENTERED INTO THE TRACK STACK WILL FOLLOW THE DATA HELD THERE.

FORM OF APPEND FOLLOWING: <APPND>

THE TRACKS DISPLAYED IN THE PREVIOUS DISPLAY WILL BE AUTOMATICALLY INCLUDED IN THE FOLLOWING DISPLAYS UNTIL CANCELLED BY <NOAPP>. ALL APPENDED TRACKS WILL NOT BE AFECTED BY A CHANGE OF THE TIME WINDOW.

FORM OF CANCEL APPEND: <NOAPP>

ALL TRACKS APPENDED BY <APPND> ARE RELEASED FROM AUTOMATIC INCLUSION TO ALL DISPLAYS.

FORM OF RE-START DISPLAY: <START>

DSPLA IS RE-INITIALIZED, SETTING ALL PARAMETERS TO DEFAULT AND ALLOWING THE RE-ENTERING OF SCREEN WIDTH AND GRID SIZE. ALL STORED SEGMENTS AND DATA DEFINED PREVIOUS TO THE EXECUTION OF THIS COMMAND ARE LOST.

FORM OF DEFINE ELLIPSE: <CELL>

ALLOWS DEFINING A CONFIDENCE ELLIPSE AND SETTING A SYMBOLIC NAME FOR THAT ELLIPSE. PARAMETERS REQUIRED ARE CENTER POSITION, MAJOR AXIS, MINOR AXIS, MAJOR AXIS BEARING, AND TIME. THE SYMBOLIC NAME MAY THEN BE USED AS THE NAME OF A TRACK WHICH PLOTS THAT ELLIPSE.

FORM OF DEFINE BEARING LINE: <LOB>

ALLOWS DEFINING A LINE OF BEARING ALONG A GREAT CIRCLE AND SETTING A SYMBOLIC NAME FOR THAT BEARING LINE. PARAMETERS REQUIRED ARE POINT OF ORIGIN AND BEARING. THE SYMBOLIC NAME MAY THEN BE USED AS THE NAME OF A TRACK WHICH PLOTS THAT BEARING LINE.

FORM OF DEFINE CIRCLE: <CIRCLE>

ALLOWS DEFINING A CIRCLE AND SETTING A SYMBOLIC NAME FOR THAT CIRCLE. PARAMETERS REQUIRED ARE CENTER POSITION AND RADIUS. THE SYMBOLIC NAME MAY THEN BE USED AS THE NAME OF A TRACK WHICH PLOTS THAT CIRCLE.

FORM OF PLOT TRACKING: <PTR><SP><UNIT CODE>, . . . ,<UNIT CODE>

A UNIT CODE MAY BE (1) THE NAME OF THE CRAFT, OF WHICH ONLY THE FIRST FIVE LETTERS NEED BE GIVEN; (2) THE TYPE OR CLASS OF CRAFT SUCH AS FFG7A; (3) THE TARGET PARAMETER OF THE CRAFT, SUCH AS FA FOR FRIENDLY AIR; (4) A FIGURE NAME DENOTING AN ELLIPSE, CIRCLE, OR LINE OF BEARING PREVIOUSLY DEFINED, SUCH AS .LOBO. UP TO 15 UNIT CODES MAY BE ENTERED IN ANY COMBINATION. THE CRAFT TRACKS ARE DRAWN FROM THE TRACKING STRUCTURE AND PLOTTED UNDER CURRENT CONDITIONS (MAP, AREA, TIME, ETC.)

FORM OF LOCATE TRACKING: <LTR><SP><UNIT CODE>, . . . ,<UNIT CODE>

A UNIT CODE MAY BE (1) THE NAME OF THE CRAFT, OF WHICH ONLY THE FIRST FIVE LETTERS NEED BE GIVEN; (2) THE TYPE OR CLASS OF CRAFT SUCH AS FFG7A; (3) THE TARGET PARAMETER OF THE CRAFT, SUCH AS FA FOR FRIENDLY AIR; (4) A FIGURE NAME DENOTING AN ELLIPSE, CIRCLE, OR LINE OF BEARING PREVIOUSLY DEFINED, SUCH AS .LOBO. UP TO 15 UNIT CODES MAY BE ENTERED IN ANY COMBINATION. THE CRAFT POSITIONS AT THE CURRENT TIME (REGARDLESS OF WHAT THE TIME WINDOW IS) ARE DRAWN FROM THE TRACKING STRUCTURE AND PLOTTED UNDER CURRENT CONDITIONS (MAP, AREA, ETC.) EXCLUDING TIME.

FORM OF LIST FILES: <FILES>

A LISTING OF ACCESSABLE FILES AND SUBFILES IS GIVEN ALONG WITH THEIR CORRESPONDING FILE CODES. TO VIEW A PARTICULAR FILE (SUBFILES ARE ACCESSABLE ONLY FROM ITS PARENT FILE) ENTER ITS CORRESPONDING FILE CODE WHEN PROMPTED.

FORM OF SET TYPE SIZE 1: <SIZE1>

FOR A TEKTRONIX 4014 TERMINAL, THE LARGEST TYPE SIZE IS SELECTED. THIS IS THE DEFAULT SIZE TYPE AND MUST BE USED ON SINGLE TYPE SIZE TERMINALS.

FORM OF SET TYPE SIZE 2: <SIZE2>

FOR A TEKTRONIX 4014 TERMINAL, THE NEXT TO LARGEST TYPE SIZE IS SELECTED. ONLY TYPE SIZE 1 SHOULD BE USED ON SINGLE TYPE SIZE TERMINALS.

FORM OF SET TYPE SIZE 3: <SIZE3>

FOR A TEKTRONIX 4014 TERMINAL, THE NEXT TO SMALLEST TYPE SIZE IS SELECTED. ONLY TYPE SIZE 1 SHOULD BE USED ON SINGLE TYPE SIZE TERMINALS.

FORM OF SET TYPE SIZE 4: <SIZE4>

FOR A TEKTRONIX 4014 TERMINAL, THE SMALLEST TYPE SIZE IS SELECTED. ONLY TYPE SIZE 1 SHOULD BE USED ON SINGLE TYPE SIZE TERMINALS.

FORM OF SET LINES PER PAGE: <PGSIZ>

ALLOWS CHANGING OF THE NUMBER OF LINES PER PAGE WHEN VIEWING FILES. THE DEFAULT PAGE SIZE IS 20 LINES. NOTE THAT MAXIMUM PAGE SIZE VARIES WITH TYPE SIZE ON MULTIPLE TYPE SIZE TERMINALS.

FORM OF HELP: <?>

GENERAL INSTRUCTIONS FOR COMMAND INPUT, FORMAT, AND OBTAINING ADDITIONAL HELP ARE GIVEN ALONG WITH A LISTING OF ALL COMMANDS AND THEIR FUNCTIONS.

FORM FOR SET GEOGRAPHIC AREA: <C><SP><MAP CODE>

MAP CODE IS THE CODE OF A MAP AREA AS DEFINED BY DSPLA. A LISTING OF AVAILABLE MAP AREAS AND THEIR CODES MAY BE OBTAINED BY EXECUTING <C><SP><LIST>. THE GLOBAL AREA WINDOW IS SET TO THE BOUNDARIES OF THE MAP BUT LAND MASS OUTLINES ARE NOT INCLUDED IN FUTURE DISPLAYS.

FORM FOR REPEAT LAST PLOT: <SP>

THE PREVIOUS PLOT OR LOCATE COMMAND IS RE-EXECUTED UNDER THE CURRENT CONDITIONS (MAP, AREA, TIME, ETC.).

FORM OF REDUCE DISPLAY: <R>

THE PREVIOUS DISPLAY WILL BE REDUCED BY THE FACTOR SET BY <SCALE>. THE DEFAULT FACTOR IS 2.

FORM OF ENQUIRE: <E>

ENQUIRE PROVIDES A DISPLAY OF STATUS, WINDOW, AND DATA RANGE INFORMATION.

FORM OF KILL DISPLAY: <K><SP><DISPLAY NAME>

DISPLAY NAME IS THE 5 CHARACTER (MAX) NAME UNDER WHICH THE DISPLAY WAS STORED. STORAGE OCCUPIED BY THE DISPLAY IS RELEASED AS IS THE DISPLAY NAME. IN CASE OF DUPLICATE NAMES, THE FIRST NAME IS DELETED.

FORM OF RECALL DISPLAY: <D><SP><DISPLAY NAME>

DISPLAY NAME IS THE 5 CHARACTER (MAX) NAME UNDER WHICH THE DISPLAY WAS STORED. THE STORED DISPLAY IS DISPLAYED ON RECALL BUT MAY NOT BE OPERATED ON (MAGNIFIED, MOVE TIME FORWARD, ETC.). IN THE CASE OF DUPLICATE NAMES, THE FIRST DISPLAY IS DISPLAYED.

FORM OF STORE DISPLAY: <S><SP><DISPLAY NAME>

DISPLAY NAME IS UP TO 5 CHARACTERS WHICH WILL IDENTIFY THE DISPLAY BEING STORED FOR LATER RECALL AND DELETION. UP TO 25 DISPLAYS MAY BE STORED. DUPLICATE NAMES SHOULD NOT BE USED.

FORM OF CHANGE TIME WINDOW: <T><SP><TIME CODE 1><TIME CODE 2>

TIME CODE 1 MAY BE (1) <I> TO ALLOW KEYBOARD ENTRY OF A NEW TIME INCREMENT (DT) - NO CODE 2 IS USED WITH THIS OPTION; (2) <G> TO DENOTE CHANGING THE GLOBAL TIME WINDOW; (3) <L> TO DENOTE CHANGING THE LOCAL TIME WINDOW. TIME CODE 2 MAY BE (1) <SP> TO DENOTE INPUT OF NEW VALUES FROM THE KEYBOARD; (2) <D> TO DENOTE DEFAULT VALUES (WHICH ARE AN INFINITE TIME WINDOW).

FORM OF CHANGE AREA WINDOW: <A><SP><AREA CODE 1><AREA CODE 2>

AREA CODE 1 MAY BE (1) <G> TO DENOTE CHANGING THE GLOBAL AREA WINDOW; (2) <L> TO DENOTE CHANGING THE LOCAL AREA WINDOW. AREA CODE 2 MAY BE (1) <SP> TO DENOTE INPUT OF NEW VALUES FROM THE KEYBOARD; (2) <C> TO DENOTE USING THE WINDOW VALUES USED OR COMPUTED IN THE PREVIOUS DISPLAY; (3) <D> TO DENOTE DEFAULT VALUES (WHICH ARE ALL ZERO) AND CAUSE AN AREA WINDOW TO BE COMPUTED TO INCLUDE ALL DATA BEING PLOTTED.

FORM OF MOVE TIME FORWARD: <F>

THE TIME WINDOW IS MOVED AHEAD BY AN INCREMENT, DT (SET BY <T><SP><I>), AND THE PREVIOUS PLOT COMMAND IS EXECUTED (<SP>).

FORM OF MAGNIFY DISPLAY: <M><SP><GRID CODE>

THE PORTION OF THE PREVIOUS DISPLAY DEFINED BY THE GRID CODE IS MAGNIFIED ACCORDING TO THE FACTOR SET BY <SCALE>. THE DEFAULT FACTOR IS 2. GRID CODES ARE DEFINED AS FOLLOWS:

1 - UPPER RIGHT QUAD	4 - LOWER RIGHT QUAD	7 - LEFT HALF
2 - UPPER LEFT QUAD	5 - CENTER	8 - LOWER HALF
3 - LOWER LEFT QUAD	6 - TOP HALF	9 - RIGHT HALF

FORM OF SET GEOGRAPHIC MAP: <G><SP><MAP CODE>

MAP CODE IS THE CODE OF A MAP AREA AS DEFINED BY DSPLA. A LISTING OF AVAILABLE MAP AREAS AND THEIR CODES MAY BE OBTAINED BY EXECUTING <G><SP><LIST>. THE GLOBAL AREA WINDOW IS SET TO THE BOUNDARIES OF THE MAP AND LAND MASS OUTLINES ARE INCLUDED IN FUTURE DISPLAYS IN MERCATOR PROJECTION FORMAT.

FORM OF LOCATE GROUND TRUTH: <LGT><SP><UNIT CODE>, . . . ,<UNIT CODE>

A UNIT CODE MAY BE (1) THE NAME OF THE CRAFT, OF WHICH ONLY THE FIRST FIVE LETTERS NEED BE GIVEN; (2) THE TYPE OR CLASS OF CRAFT SUCH AS FFG7A; (3) THE TARGET PARAMETER OF THE CRAFT, SUCH AS FA FOR FRIENDLY AIR; (4) A FIGURE NAME DENOTING AN ELLIPSE, CIRCLE, OR LINE OF BEARING PREVIOUSLY DEFINED, SUCH AS .LOBO. UP TO 15 UNIT CODES MAY BE ENTERED IN ANY COMBINATION. THE CRAFT POSITIONS AT THE CURRENT TIME (REGARDLESS OF WHAT THE TIME WINDOW IS) ARE DRAWN FROM THE GROUND TRUTH STRUCTURE AND PLOTTED UNDER CURRENT CONDITIONS (MAP, AREA, ETC.) EXCLUDING TIME.

FORM OF QUIT: <Q>

QUIT CAUSES A RETURN TO THE CALLING PROGRAM.

FORM OF PLOT GROUND TRUTH: <PGT><SP><UNIT CODE>, . . . ,<UNIT CODE>

A UNIT CODE MAY BE (1) THE NAME OF THE CRAFT, OF WHICH ONLY THE FIRST FIVE LETTERS NEED BE GIVEN; (2) THE TYPE OR CLASS OF CRAFT SUCH AS FFG7A; (3) THE TARGET PARAMETER OF THE CRAFT, SUCH AS FA FOR FRIENDLY AIR; (4) A FIGURE NAME DENOTING AN ELLIPSE, CIRCLE, OR LINE OF BEARING PREVIOUSLY DEFINED, SUCH AS .LOBO. UP TO 15 UNIT CODES MAY BE ENTERED IN ANY COMBINATION. THE CRAFT TRACKS ARE DRAWN FROM THE GROUND TRUTH STRUCTURE AND PLOTTED UNDER CURRENT CONDITIONS (MAP, AREA, TIME, ETC.)

INDEX OF POSITION - DSPLAE

ENTER ONE OF THE ABOVE POSITION INDICIES WHICH IS TO BE USED AS A SOURCE FOR THE FIGURE CENTER OR ORIGIN. THE POSITION MOST RECENTLY ENTERED IN THE CURSOR STACK APPEARS FIRST, THE SECOND MOST RECENT SECOND, ETC.

LONGITUDE OF CENTER OR ORIGIN - DSPLAE

ENTER A LONGITUDE IN DEGREES AND MINUTES WITH DEGREES SEPARATED FROM MINUTES BY A MINUS, '-', AND WEST LONGITUDE DENOTED BY A LEADING MINUS SIGN.

LATITUDE OF CENTER OR ORIGIN - DSPLAE

ENTER A LATITUDE IN DEGREES AND MINUTES WITH DEGREES SEPARATED FROM MINUTES BY A MINUS, '-', AND SOUTH LATITUDE DENOTED BY A LEADING MINUS SIGN.

SOURCE OF INPUT OF CENTER OR ORIGIN - DSPLAE

ENTER 'C' TO DENOTE THE CURSOR STACK AS THE SOURCE OF LAT-LON POSITIONS; 'K' TO DENOTE INPUT OF LAT-LON FROM THE KEYBOARD. THIS POSITION IS USED AS THE FIGURE CENTER OR ORIGIN.

SPEED FOR RADIUS - DSPLAE

ENTER A SPEED AS AN INTEGRAL NUMBER OF KNOTS. THIS SPEED IS MULTIPLIED BY THE CURRENT VALUE OF THE TIME INCREMENT, DT, TO ARRIVE AT A DISTANCE WHICH IS USED AS THE RADIUS FOR THE CIRCLE.

BEARING FOR RHUMB LINE - DSPLAE

ENTER THE BEARING OF THE RHUMB LINE AS POSITIVE MINUTES AND DEGREES FROM TRUE NORTH. MINUTES ARE SEPARATED FROM DEGREES BY A MINUS, '-'.

RADIUS OF CIRCLE - DSPLAE

ENTER THE RADIUS OF THE CIRCLE AS AN INTEGRAL NUMBER OF MILES.

BEARING OF GREAT CIRCLE LINE - DSPLAE

ENTER THE BEARING OF THE GREAT CIRCLE LINE AS POSITIVE MINUTES AND DEGREES FROM TRUE NORTH. MINUTES ARE SEPARATED FROM DEGREES BY A MINUS, '-'.

TIME OF ELLIPSE - DSPLAE

ENTER AN INTEGRAL TIME VALUE IN MINUTES TO BE ASSOCIATED WITH THE ELLIPSE.

BEARING OF ELLIPSE - DSPLAE

ENTER THE BEARING OF THE MAJOR AXIS OF THE ELLIPSE AS POSITIVE DEGREES AND MINUTES FROM TRUE NORTH. MINUTES ARE SEPARATED FROM DEGREES BY A MINUS, '-'.

LENGTH OF MINOR AXIS OF ELLIPSE - DSPLAE

ENTER THE LENGTH OF THE MINOR AXIS OF THE ELLIPSE AS AN INTEGRAL NUMBER OF MILES.

LENGTH OF MAJOR AXIS OF ELLIPSE - DSPLAE

ENTER THE LENGTH OF THE MAJOR AXIS OF THE ELLIPSE AS AN INTEGRAL NUMBER OF MILES.

GDXLON - DSPLAA

ENTER A POSITIVE LONGITUDE INCREMENT IN DEGREES AND MINUTES WHICH IS TO REPRESENT ONE FOURTH OF THE LONGITUDE SPAN ACROSS THE GLOBAL AREA WINDOW. MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-'. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. GXOLON + 4(GDXLON) THEN GIVES THE ENDING VALUE (RIGHT BOUNDARY) OF THE GLOBAL AREA WINDOW. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE GLOBAL AREA WINDOW WILL NOT BE DISPLAYED. THE GLOBAL AREA WINDOW IS USED AT ALL TIMES EXCEPT FOLLOWING THE SETTING OF THE LOCAL AREA WINDOW.

GDYLAT - DSPLAA

ENTER A POSITIVE LATITUDE INCREMENT IN DEGREES AND MINUTES WHICH IS TO REPRESENT ONE FOURTH OF THE LATITUDE SPAN ACROSS THE GLOBAL AREA WINDOW. MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-'. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. GYOLAT + 4(GDYLAT) THEN GIVES THE ENDING VALUE (UPPER BOUNDARY) OF THE GLOBAL AREA WINDOW. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE GLOBAL AREA WINDOW WILL NOT BE DISPLAYED. THE GLOBAL AREA WINDOW IS USED AT ALL TIMES EXCEPT FOLLOWING THE SETTING OF THE LOCAL AREA WINDOW.

DYLAT - DSPLAA

ENTER A POSITIVE LATITUDE INCREMENT IN DEGREES AND MINUTES WHICH IS TO REPRESENT ONE FOURTH OF THE LATITUDE SPAN ACROSS THE LOCAL AREA WINDOW. MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-'. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. YOLAT + 4(DYLAT) THEN GIVES THE ENDING VALUE (UPPER BOUNDARY) OF THE LOCAL AREA WINDOW. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE LOCAL AREA WINDOW WILL NOT BE DISPLAYED. ONCE THE LOCAL AREA WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

DXLON - DSPLAA

ENTER A POSITIVE LONGITUDE INCREMENT IN DEGREES AND MINUTES WHICH IS TO REPRESENT ONE FOURTH OF THE LONGITUDE SPAN ACROSS THE LOCAL AREA WINDOW. MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-'. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. XOLON + 4(DXLON) THEN GIVES THE ENDING VALUE (RIGHT BOUNDARY) OF THE LOCAL AREA WINDOW. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE LOCAL AREA WINDOW WILL NOT BE DISPLAYED. ONCE THE LOCAL AREA WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

GXOLON - DSPLAA

ENTER A LONGITUDE IN DEGREES AND MINUTES AT WHICH THE GLOBAL AREA WINDOW WILL BEGIN (LEFT BOUNDARY). MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-', AND WEST LONGITUDES ARE DENOTED BY A LEADING MINUS SIGN. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE GLOBAL AREA WINDOW WILL NOT BE DISPLAYED. THE GLOBAL AREA WINDOW IS USED AT ALL TIMES EXCEPT FOLLOWING THE SETTING OF THE LOCAL AREA WINDOW.

GYOLAT - DSPLAA

ENTER A LATITUDE IN DEGREES AND MINUTES AT WHICH THE GLOBAL AREA WINDOW WILL BEGIN (LOWER BOUNDARY). MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-', AND SOUTH LATITUDES ARE DENOTED BY A LEADING MINUS SIGN. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE GLOBAL AREA WINDOW WILL NOT BE DISPLAYED. THE GLOBAL AREA WINDOW IS USED AT ALL TIMES EXCEPT FOLLOWING THE SETTING OF THE LOCAL AREA WINDOW.

YOLAT - DSPLAA

ENTER A LATITUDE IN DEGREES AND MINUTES AT WHICH THE LOCAL AREA WINDOW WILL BEGIN (LOWER BOUNDARY). MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-', AND SOUTH LATITUDES ARE DENOTED BY A LEADING MINUS SIGN. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE LOCAL AREA WINDOW WILL NOT BE DISPLAYED. ONCE THE LOCAL AREA WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

XOLON - DSPLAA

ENTER A LONGITUDE IN DEGREES AND MINUTES AT WHICH THE LOCAL AREA WINDOW WILL BEGIN (LEFT BOUNDARY). MINUTES AND DEGREES ARE SEPARATED BY A MINUS, '-', AND WEST LONGITUDES ARE DENOTED BY A LEADING MINUS SIGN. MINUTES NEED NOT BE ENTERED IF ZERO, HOWEVER, IF DEGREES ARE ZERO, A ZERO MUST BE ENTERED. THE TRACKS OR PARTS OF TRACKS THAT OCCUR IN AREAS OUTSIDE THE LOCAL AREA WINDOW WILL NOT BE DISPLAYED. ONCE THE LOCAL AREA WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

GTIMEF - DSPLAT

ENTER AN INTEGRAL TIME IN MINUTES THAT THE GLOBAL TIME WINDOW IS TO END AT. TRACKS OR PARTS OF TRACKS THAT OCCUR AT TIMES AFTER GTIMEF WILL NOT BE DISPLAYED (UNLESS THE LOCATE COMMAND IS USED). THE GLOBAL TIME WINDOW IS USED AT ALL TIMES EXCEPT THE FIRST DISPLAY FOLLOWING THE SETTING OF THE LOCAL TIME WINDOW.

GTIMEI - DSPLAT

ENTER AN INTEGRAL TIME IN MINUTES THAT THE GLOBAL TIME WINDOW IS TO BEGIN AT. TRACKS OR PARTS OF TRACKS THAT OCCUR AT TIMES BEFORE GTIMEI WILL NOT BE DISPLAYED (UNLESS THE LOCATE COMMAND IS USED). THE GLOBAL TIME WINDOW IS USED AT ALL TIMES EXCEPT THE FIRST DISPLAY FOLLOWING THE SETTING OF THE LOCAL TIME WINDOW.

TIMEF - DSPLAT

ENTER AN INTEGRAL TIME IN MINUTES THAT THE LOCAL TIME WINDOW IS TO END AT. TRACKS OR PARTS OF TRACKS THAT OCCUR AT TIMES AFTER TIMEF WILL NOT BE DISPLAYED (UNLESS THE LOCATE COMMAND IS USED) IF THE LOCAL TIME WINDOW IS USED. ONCE THE LOCAL TIME WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

TIMEI - DSPLAT

ENTER AN INTEGRAL TIME IN MINUTES THAT THE LOCAL TIME WINDOW IS TO BEGIN AT. TRACKS OR PARTS OF TRACKS THAT OCCUR AT TIMES BEFORE TIMEI WILL NOT BE DISPLAYED (UNLESS THE LOCATE COMMAND IS USED) IF THE LOCAL TIME WINDOW IS USED. ONCE THE LOCAL TIME WINDOW IS SET, IT AFFECTS ONLY THE FIRST FOLLOWING DISPLAY.

DT - DSPLAT

ENTER AN INTEGRAL TIME INCREMENT (DT) IN MINUTES. DT IS USED IN A NUMBER OF PLACES. WHEN THE 'F' AND 'B' COMMANDS OR FUNCTION KEYS ARE USED, DT IS THE AMOUNT THAT THE TIME WINDOW IS MOVED FORWARD OR BACKWARD. DT IS USED IN DEFINING A RADIUS; THE SPEED IS MULTIPLIED BY DT TO GIVE A RADIUS OF THE RESULTING CIRCLE. DT IS THE TIME INCREMENT BETWEEN POINTS OF A TRACK GENERATED BY THE 'K' FUNCTION KEY.

SPEED FOR RANGE, BEARING, TIME - DSPLA

ENTER AN INTEGRAL SPEED IN KNOTS TO BE USED AS THE SPEED OF THE CRAFT FOR CALCULATING TIME OF TRAVEL BETWEEN THE POINTS OF THE TRACK IN THE TRACK STACK.

TRACK NUMBER - DSPLA

ENTER A 3 DIGIT INTEGER WHICH WILL BECOME THE TRACK NUMBER OF THE TRACK. THE NUMBER SHOULD BE UNIQUE. TRACK NUMBERS ARE LISTED ALONG WITH OTHER TRACK HEADER INFORMATION WHEN DISPLAYING THE TRACK FILE. AN EDITING FEATURE IS AVAILABLE FOR CHANGING THE HEADER AND TRACK DATA WHEN DISPLAYING THE TRACK FILE.

TRACK ALTITUDE - DSPLA

ENTER THE ALTITUDE OF THE CRAFT GENERATING THE TRACK. THE ALTITUDE SHOULD BE AN INTEGRAL NUMBER OF FEET AND WILL BE KEPT CONSTANT THROUGH TRACK GENERATION.

INITIAL TRACK TIME - DSPLA

ENTER THE TIME OF THE INITIAL POINT OF THE TRACK AS AN INTEGRAL NUMBER OF MINUTES. SUCCEEDING POINTS WILL BE A TIME 'DT' AFTER THE PRECEEDING POINT. DT (TIME INCREMENT) MUST BE SET PRIOR TO GENERATION OF A TRACK AND IS CHANGED BY COMMAND.

SCALE - DSPLA

ENTER THE FACTOR BY WHICH DISPLAYS ARE MAGNIFIED OR REDUCED AS THE NUMBER OF DEGREES AND MINUTES WHICH ARE TO CORRESPOND TO 1-00 DEGREES IN THE OLD DISPLAY. DEGREES AND MINUTES ARE TO BE SEPARATED BY A MINUS, '-'. THE DISPLAY WILL BE MAGNIFIED OR REDUCED BY A FACTOR OF THE NEW SCALE VALUE WHENEVER A MAGNIFY OR REDUCE COMMAND IS EXECUTED.

NUMBER OF LINES PER PAGE - DSPLA

ENTER THE NUMBER OF LINES PER PAGE FOR FILE DISPLAY AS AN INTEGER NUMBER. DEFAULT PAGE SIZE IS 20 LINES. NOTE THAT AS THE TYPE SIZE IS DECREASED, THE MAXIMUM NUMBER OF LINES PER PAGE INCREASES, AND VICA VERSA.

FILE CODE - DSPLA

TO VIEW ONE OF THE ABOVE LISTED FILES, ENTER THE CORRESPONDING FILE CODE. SUBFILE CODES MAY NOT BE ENTERED EXCEPT WITHIN THE VIEWING OF ITS PARENT FILE. ONCE THE FILE IS DISPLAYED, USE 'F' TO VIEW SUCCEEDING PAGES; 'B' TO VIEW PRECEEDING PAGES; 'R' TO RETURN FROM VIEWING THAT PAGE, 'E' TO EDIT A LINE; 'Q' TO QUIT VIEWING; 'C' TO RETURN TO COMMAND MODE IN DSPLA; AND ANY SUBFILE CODES TO VIEW SUBFILES. TYPE SIZE MAY BE CHANGED BY COMMAND OR FUNCTION KEY. TEKTRONIX 4012 TERMINALS MUST USE TYPE SIZE1.

Appendix B
Memory Structure

B-1

The following is a list of the most important relations and basenode types in the demonstration system. The format used is <BASENODE TYPE> <RELATION which descends from that type>*.

PLATFORMS have NAME, CLASS, CATEGORY, TYPE, MEDIUM, MAX-SPEED, CRUISE-SPEED, HOSTILITY, WARLIKE, SIGHTINGs, FIRST-SIGHTING, LAST-SIGHTING.

SIGHTINGs have LATITUDE, LONGITUDE, TIME, COURSE, SPEED, RANGE, BEARING, SOURCE, MESSAGE, SUCCESSOR

STORMs have VERTICES

MERCHANTLANEs have VERTICES, START, DESTINATION, TO-PORT, FROM-PORT

VERTICES have LATITUDE, LONGITUDE

The technical data base uses relations much like those of PLATFORMs, with the following changes: there are no sightings, and all relation names are prefaced with T, e.g. TYPE becomes TTYPE.

These are the oracles that are defined for the demonstration system.

```
(SAME-AS
  [LAMBDA (W U)
    (EQ W U)])

(REACHABLE-BY-A-COMBATANT
  [LAMBDA (FSBLIP DMY)                                    (* NOBIND "15-Nov-78 10:09")
    (PROG (YES)
          [MAPC COMBATANTS (FUNCTION (LAMBDA (CMBT)
                    (PROG (SITE)
                          (CASSERT CMBT (QUOTE % )
                                   (QUOTE COMBATANT)
                                   1.0)
                          (SETQ SITE (CAAR (RETRIEVE2 (QUOTE LAST-SIGHTING)
                                                      CMBT)))
                          (COND
                            ((WITHIN-REACH FSBLIP SITE)
                              (CASSERT FSBLIP (QUOTE WITHIN-REACH)
                                       SITE 1.0)
                              (SETQ YES T))
                            (T (CASSERT FSBLIP (QUOTE WITHIN-REACH)
                                       SITE -1.0]
          (COND
            (YES (RETURN T])

(ROUGHLY-THE-SAME-SPEED-AS
  [LAMBDA (Q1 Q2)
    (AND (LESSP (MASSAGE Q1)
                (PLUS (MASSAGE Q2)
                      1.5))
         (GREATERP (MASSAGE Q1)
                   (DIFFERENCE (MASSAGE Q2)
                               1.5])

(ROUGHLY-THE-SAME-COURSE-AS
  [LAMBDA (Q1 Q2)                                         (* NOBIND " 6-Dec-78 18:49")
    (AND (LESSP (MASSAGE Q1)
                (PLUS (MASSAGE Q2)
                      4.5))
         (GREATERP (MASSAGE Q1)
                   (DIFFERENCE (MASSAGE Q2)
                               4.5])
```

```
(IN-LANE
  [LAMBDA (POS MLANE)
    (FGREATERP 5.0 (LANERANGE [EVAL (CAAR (RETRIEVE2
                                            (QUOTE LATITUDE)
                                            (CAAR (RETRIEVE2 (QUOTE START)
                                                             MLANE]
                                  [EVAL (CAAR (RETRIEVE2
                                            (QUOTE LONGITUDE)
                                            (CAAR (RETRIEVE2 (QUOTE START)
                                                             MLANE]
                                  [EVAL (CAAR (RETRIEVE2
                                            (QUOTE LATITUDE)
                                            (CAAR (RETRIEVE2 (QUOTE DESTINATION
                                                             MLANE]
                                  [EVAL (CAAR (RETRIEVE2
                                            (QUOTE LONGITUDE)
                                            (CAAR (RETRIEVE2 (QUOTE DESTINATION
                                                             MLANE]
                                  (EVAL (CAAR (RETRIEVE2 (QUOTE LATITUDE)
                                                         POS)))
                                  (EVAL (CAAR (RETRIEVE2 (QUOTE LONGITUDE)
                                                         POS])

(INSIDE
  [LAMBDA (POS STORM)
    (APPLY (QUOTE INTERIOR)
           (APPEND (GETPOS POS)
                   (CONS (GETVERS STORM])

(GREATER-THAN
  [LAMBDA (Q1 Q2)
    (GREATERP (MASSAGE Q1)
              (MASSAGE Q2])

(LESS-THAN
  [LAMBDA (Q1 Q2)
    (LESSP (MASSAGE Q1)
           (MASSAGE Q2])

(WITHIN-REACH
  [LAMBDA (S1 S2)
    (APPLYRULE (QUOTE GOOD-PATH])
```

```
(BLOCKEDFROM
  [LAMBDA (S1 S2)

          (* Tests if some patrol report is inconsistent with a
          passage from S1 to S2)


    (PROG (SUCCESSFLG)
          [MAPC
            PATROLS
            (FUNCTION (LAMBDA (PTL)
                (COND
                  ((NOT (POSS-REPORT S1 S2 PTL))
                    (MAPC
                      (RETRIEVE2 (QUOTE SIGHTING)
                                 PTL)
                      (FUNCTION (LAMBDA (SNG-AS)
                          (PROG ((SNG (CAR SNG-AS)))
                                (MAPC
                                  (RETRIEVE2 (QUOTE SUCCESSOR)
                                             SNG)
                                  (FUNCTION (LAMBDA (SNG2-AS)
                                      (PROG ((SNG2 (CAR SNG2-AS)))
                                            (COND
                                              ((OR (CROSSPATHS S1 S2 SNG SNG2)
                                                   (GRAZE S1 S2 SNG SNG2))
                                                (COND
                                                  ((NOT (WENT-BEFORE S1 S2 SNG
                                                                     SNG2))
                                                    (COND
                                                      ((NOT (WENT-AFTER S1 S2
                                                                        SNG
                                                                        SNG2))
                                                        (SETQ SUCCESSFLG T]
          (RETURN SUCCESSFLG])

(SIMPLY-WITHIN-REACH
  [LAMBDA (S1 S2)
    (PROG ((LT1 (LOC-TIME S1))
           (LT2 (LOC-TIME S2)))
          (RETURN (SWR LT1 LT2])
```

Appendix C

The following is the contents of memory at the start of the demonstration scenario.

```
SURFACE is the TMEDIUM of CV
COMBATANT is the TAGGRESSIVENESS of CV
NO223 is the SUCCESSOR of NO222
AIR is the TMEDIUM of FIGHTER
480 is the TIME of NO220
COMBATANT is the TAGGRESSIVENESS of FIGHTER
MILITARY is the TTYPE of FIGHTER
US is the TFLAG of F16-A
FRIENDLY is the THOSTILITY of F16-A
FIGHTER is the TCATEGORY of F16-A
F16-A is the TCLASS of HUMMER
SURFACE is the TMEDIUM of MERCHANT
SURFACE is the TMEDIUM of CLG
-27.54 is the LONGITUDE of NO225
62.37 is the LATITUDE of NO225
-27.06 is the LONGITUDE of NO224
62.49 is the LATITUDE of NO224
-26.66 is the LONGITUDE of NO223
62.68 is the LATITUDE of NO223
-26.2 is the LONGITUDE of NO222
62.84 is the LATITUDE of NO222
-25.69 is the LONGITUDE of NO221
62.98 is the LATITUDE of NO221
-25.13 is the LONGITUDE of NO220
63.05 is the LATITUDE of NO220
-24.61 is the LONGITUDE of NO219
63.19 is the LATITUDE of NO219
-24.09 is the LONGITUDE of NO218
63.36 is the LATITUDE of NO218
-23.73 is the LONGITUDE of NO217
63.57 is the LATITUDE of NO217
-23.43 is the LONGITUDE of NO216
63.81 is the LATITUDE of NO216
-23.38 is the LONGITUDE of NO215
63.99 is the LATITUDE of NO215
780 is the TIME of NO225
720 is the TIME of NO224
660 is the TIME of NO223
600 is the TIME of NO222
540 is the TIME of NO221
420 is the TIME of NO219
360 is the TIME of NO218
300 is the TIME of NO217
240 is the TIME of NO216
180 is the TIME of NO215
NO225 is the LAST-SIGHTING of PERRY
NO225 is the SUCCESSOR of NO224
NO224 is the SUCCESSOR of NO223
NO222 is the SUCCESSOR of NO221
NO221 is the SUCCESSOR of NO220
```

```
NO220 is the SUCCESSOR of NO219
NO219 is the SUCCESSOR of NO218
NO218 is the SUCCESSOR of NO217
NO217 is the SUCCESSOR of NO216
NO216 is the SUCCESSOR of NO215
NO215 is the SUCCESSOR of SIGHTING0205
-23.38 is the LONGITUDE of SIGHTING0205
63.99 is the LATITUDE of SIGHTING0205
0.0 is the RANGE of SIGHTING0205
0.0 is the BEARING of SIGHTING0205
INTERNAL is the SOURCE of SIGHTING0205
0.0 is the TIME of SIGHTING0205
SIGHTING0205 is the FIRST-SIGHTING of PERRY
SIGHTING0205 is the LAST-SIGHTING of PERRY
SIGHTING0205 is a SIGHTING of PERRY
PERRY is the NAME of PERRY
PERRY is the CLASS of PERRY
FFG is the CATEGORY of PERRY
30.0 is the MAX-SPEED of PERRY
15.0 is the CRUISE-SPEED of PERRY
NAVY is the TYPE of PERRY
SURFACE is the MEDIUM of PERRY
FRIENDLY is the HOSTILITY of PERRY
COMBATANT is the WARLIKE of PERRY
US is the FLAG of PERRY
CV is the TCATEGORY of KIEV
CLG is the TCATEGORY of KYNDA
CG is the TCATEGORY of KRESTA
FF is the TCATEGORY of KNOX
CV is the TCATEGORY of KITTYHAWK
AO is the TCATEGORY of KAZBEK
DDC is the TCATEGORY of KASHIN
30.0 is the TMAX-SPEED of KIEV
16.0 is the TCRUISE-SPEED of KIEV
UR is the TFLAG of KIEV
HOSTILE is the THOSTILITY of KIEV
35.0 is the TMAX-SPEED of KYNDA
17.5 is the TCRUISE-SPEED of KYNDA
UR is the TFLAG of KYNDA
HOSTILE is the THOSTILITY of KYNDA
33.0 is the TMAX-SPEED of KRESTA
16.5 is the TCRUISE-SPEED of KRESTA
UR is the TFLAG of KRESTA
HOSTILE is the THOSTILITY of KRESTA
27.0 is the TMAX-SPEED of KNOX
16.0 is the TCRUISE-SPEED of KNOX
US is the TFLAG of KNOX
FRIENDLY is the THOSTILITY of KNOX
35.0 is the TMAX-SPEED of KITTYHAWK
16.0 is the TCRUISE-SPEED of KITTYHAWK
```

```
US is the TFLAG of KITTYHAWK
FRIENDLY is the THOSTILITY of KITTYHAWK
14.0 is the TMAX-SPEED of KAZBEK
10.0 is the TCRUISE-SPEED of KAZBEK
UR is the TFLAG of KAZBEK
HOSTILE is the THOSTILITY of KAZBEK
35.0 is the TMAX-SPEED of KASHIN
18.0 is the TCRUISE-SPEED of KASHIN
UR is the TFLAG of KASHIN
HOSTILE is the THOSTILITY of KASHIN
NON-COMBATANT is the TAGGRESSIVENESS of MERCHANT
CIVILIAN is the TTYPE of MERCHANT
COMBATANT is the TAGGRESSIVENESS of CG
NAVY is the TTYPE of CG
NAVY is the TTYPE of CV
COMBATANT is the TAGGRESSIVENESS of AO
NAVY is the TTYPE of AO
COMBATANT is the TAGGRESSIVENESS of CLG
NAVY is the TTYPE of CLG
COMBATANT is the TAGGRESSIVENESS of FF
NAVY is the TTYPE of FF
COMBATANT is the TAGGRESSIVENESS of DDC
NAVY is the TTYPE of DDC
KIEV is the TCLASS of MINSK
KYNDA is the TCLASS of ADMIRAL-GOLOVKO
KRESTA is the TCLASS of ADMIRAL-MAKAROV
KNOX is the TCLASS of RATHBURNE
KNOX is the TCLASS of MEYERCORD
KNOX is the TCLASS of CONNOLE
KITTYHAWK is the TCLASS of CONSTELLATION
KAZBEK is the TCLASS of DESNA
KASHIN is the TCLASS of PROVORNY
VERTEX0131 is the DESTINATION of LANE3
VERTEX0131 is the SUCCESSOR of VERTEX0126
-6.72 is the LONGITUDE of VERTEX0131
63.79 is the LATITUDE of VERTEX0131
VERTEX0131 is a VERTEX of LANE3
VERTEX0126 is the SUCCESSOR of VERTEX0121
-13.96 is the LONGITUDE of VERTEX0126
62.99 is the LATITUDE of VERTEX0126
VERTEX0126 is a VERTEX of LANE3
VERTEX0121 is the SUCCESSOR of VERTEX0116
-19.37 is the LONGITUDE of VERTEX0121
62.08 is the LATITUDE of VERTEX0121
VERTEX0121 is a VERTEX of LANE3
VERTEX0116 is the SUCCESSOR of VERTEX0111
-23.79 is the LONGITUDE of VERTEX0116
61.17 is the LATITUDE of VERTEX0116
VERTEX0116 is a VERTEX of LANE3
VERTEX0111 is the SUCCESSOR of VERTEX0106
```

-29.01 is the LONGITUDE of VERTEX0111
59.77 is the LATITUDE of VERTEX0111
VERTEX0111 is a VERTEX of LANE3
VERTEX0106 is the SUCCESSOR of VERTEX0101
-32.89 is the LONGITUDE of VERTEX0106
58.56 is the LATITUDE of VERTEX0106
VERTEX0106 is a VERTEX of LANE3
VERTEX0101 is the SUCCESSOR of VERTEX0096
-36.36 is the LONGITUDE of VERTEX0101
57.23 is the LATITUDE of VERTEX0101
VERTEX0101 is a VERTEX of LANE3
VERTEX0096 is the START of LANE3
-39.84 is the LONGITUDE of VERTEX0096
55.66 is the LATITUDE of VERTEX0096
VERTEX0096 is a VERTEX of LANE3
MURMANSK is the TO-PORT of LANE3
ST.JOHNS is the FROM-PORT of LANE3
VERTEX0088 is the DESTINATION of LANE2
VERTEX0088 is the SUCCESSOR of VERTEX0083
-22.99 is the LONGITUDE of VERTEX0088
64.01 is the LATITUDE of VERTEX0088
VERTEX0088 is a VERTEX of LANE2
VERTEX0083 is the SUCCESSOR of VERTEX0078
-26.0 is the LONGITUDE of VERTEX0083
63.19 is the LATITUDE of VERTEX0083
VERTEX0083 is a VERTEX of LANE2
VERTEX0078 is the SUCCESSOR of VERTEX0073
-29.94 is the LONGITUDE of VERTEX0078
61.85 is the LATITUDE of VERTEX0078
VERTEX0078 is a VERTEX of LANE2
VERTEX0073 is the SUCCESSOR of VERTEX0068
-33.75 is the LONGITUDE of VERTEX0073
60.37 is the LATITUDE of VERTEX0073
VERTEX0073 is a VERTEX of LANE2
VERTEX0068 is the SUCCESSOR of VERTEX0063
-37.9 is the LONGITUDE of VERTEX0068
58.45 is the LATITUDE of VERTEX0068
VERTEX0068 is a VERTEX of LANE2
VERTEX0063 is the START of LANE2
-42.25 is the LONGITUDE of VERTEX0063
56.04 is the LATITUDE of VERTEX0063
VERTEX0063 is a VERTEX of LANE2
REYKJAVIK is the TO-PORT of LANE2
ST.JOHNS is the FROM-PORT of LANE2
VERTEX0055 is the DESTINATION of LANE1
VERTEX0055 is the SUCCESSOR of VERTEX0050
-41.7 is the LONGITUDE of VERTEX0055
58.14 is the LATITUDE of VERTEX0055
VERTEX0055 is a VERTEX of LANE1
VERTEX0050 is the SUCCESSOR of VERTEX0045

```
-40.16 is the LONGITUDE of VERTEX0050
59.21 is the LATITUDE of VERTEX0050
VERTEX0050 is a VERTEX of LANE1
VERTEX0045 is the SUCCESSOR of VERTEX0040
-37.76 is the LONGITUDE of VERTEX0045
60.64 is the LATITUDE of VERTEX0045
VERTEX0045 is a VERTEX of LANE1
VERTEX0040 is the SUCCESSOR of VERTEX0035
-35.08 is the LONGITUDE of VERTEX0040
62.11 is the LATITUDE of VERTEX0040
VERTEX0040 is a VERTEX of LANE1
VERTEX0035 is the SUCCESSOR of VERTEX0030
-32.47 is the LONGITUDE of VERTEX0035
63.34 is the LATITUDE of VERTEX0035
VERTEX0035 is a VERTEX of LANE1
VERTEX0030 is the SUCCESSOR of VERTEX0025
-30.47 is the LONGITUDE of VERTEX0030
64.19 is the LATITUDE of VERTEX0030
VERTEX0030 is a VERTEX of LANE1
VERTEX0025 is the SUCCESSOR of VERTEX0020
-28.53 is the LONGITUDE of VERTEX0025
65.02 is the LATITUDE of VERTEX0025
VERTEX0025 is a VERTEX of LANE1
VERTEX0020 is the SUCCESSOR of VERTEX0015
-25.32 is the LONGITUDE of VERTEX0020
66.11 is the LATITUDE of VERTEX0020
VERTEX0020 is a VERTEX of LANE1
VERTEX0015 is the SUCCESSOR of VERTEX0010
-23.11 is the LONGITUDE of VERTEX0015
66.79 is the LATITUDE of VERTEX0015
VERTEX0015 is a VERTEX of LANE1
VERTEX0010 is the SUCCESSOR of VERTEX0005
-16.57 is the LONGITUDE of VERTEX0010
68.39 is the LATITUDE of VERTEX0010
VERTEX0010 is a VERTEX of LANE1
VERTEX0005 is the START of LANE1
-13.82 is the LONGITUDE of VERTEX0005
68.93 is the LATITUDE of VERTEX0005
VERTEX0005 is a VERTEX of LANE1
REYKJAVIK is the TO-PORT of LANE1
MURMANSK is the FROM-PORT of LANE1
```

Appendix D

These are the productions used in the demonstration system.

```
NAME: ID1

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE NAME) X))
(UNLESS (RETRIEVE2 (QUOTE CLASS) X))
(GETS Z (RETRIEVE2 'TCLASS Y))

ACTION:
(ASSERT Z 'CLASS X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID2

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CLASS) X))
(UNLESS (RETRIEVE2 (QUOTE CATEGORY) X))
(GETS Z (RETRIEVE2 (QUOTE TCATEGORY) Y))

ACTION:
(ASSERT Z (QUOTE CATEGORY) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID3

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CLASS) X))
(UNLESS (RETRIEVE2 (QUOTE CRUISE-SPEED) X))
(GETS Z (RETRIEVE2 (QUOTE TCRUISE-SPEED) Y))

ACTION:
(ASSERT Z (QUOTE CRUISE-SPEED) X)

CONFIDENCE: 1.0
```



```
DESCRIPTION:
NIL
```

```
NAME: ID4

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CLASS) X))
(UNLESS (RETRIEVE2 (QUOTE MAX-SPEED) X))
(GETS Z (RETRIEVE2 (QUOTE TMAX-SPEED) Y))

ACTION:
(ASSERT Z (QUOTE MAX-SPEED) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID5

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CLASS) X))
(UNLESS (RETRIEVE2 (QUOTE FLAG) X))
(GETS Z (RETRIEVE2 (QUOTE TFLAG) Y))

ACTION:
(ASSERT Z (QUOTE FLAG) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID6

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CLASS) X))
(UNLESS (RETRIEVE2 (QUOTE HOSTILITY) X))
(GETS Z (RETRIEVE2 (QUOTE THOSTILITY) Y))

ACTION:
(ASSERT Z (QUOTE HOSTILITY) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL
```

```
NAME: ID7

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CATEGORY) X))
(UNLESS (RETRIEVE2 (QUOTE WARLIKE) X))
(GETS Z (RETRIEVE2 (QUOTE TAGGRESSIVENESS) Y))

ACTION:
(ASSERT Z (QUOTE WARLIKE) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID8

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Y (RETRIEVE2 (QUOTE CATEGORY) X))
(UNLESS (RETRIEVE2 (QUOTE TYPE) X))
(GETS Z (RETRIEVE2 (QUOTE TTYPE) Y))

ACTION:
(ASSERT Z (QUOTE TYPE) X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL


NAME: ID9

CONDITIONS:
(GETS X (RETRIEVE2 '%  'PLATFORM))
(GETS Y (RETRIEVE2 'CATEGORY X))
(UNLESS (RETRIEVE2 'MEDIUM X))
(GETS Z (RETRIEVE2 'TMEDIUM Y))

ACTION:
(ASSERT Z 'MEDIUM X)

CONFIDENCE: 1.0

DESCRIPTION:
NIL
```

```
NAME: ID-NOTWRLK

CONDITIONS:
(GETS X (RETRIEVE2 '%  'PLATFORM))
(GETS Y (RETRIEVE2 'WARLIKE X))
(GETOP Z TAGGRESSIVENESS)
(UNLESS (RETRIEVE3 Z 'SAME-AS Y))

ACTION:
(ASSERT Z 'WARLIKE X)

CONFIDENCE: -1.0

DESCRIPTION:
NIL


NAME: ID-NOTMED

CONDITIONS:
(GETS X (RETRIEVE2 '%  'PLATFORM))
(GETS Y (RETRIEVE2 'MEDIUM X))
(GETOP Z TMEDIUM)
(UNLESS (RETRIEVE3 Z 'SAME-AS Y))

ACTION:
(ASSERT Z 'MEDIUM X)

CONFIDENCE: -1.0

DESCRIPTION:
NIL


NAME: ID-NOTFLG

CONDITIONS:
(GETS X (RETRIEVE2 '%  'PLATFORM))
(GETS Y (RETRIEVE2 'FLAG X))
(GETOP Z TFLAG)
(UNLESS (RETRIEVE3 Z 'SAME-AS Y))

ACTION:
(ASSERT Z 'FLAG X)

CONFIDENCE: -1.0

DESCRIPTION:
NIL
```

```
NAME: ID-NOTHOST

CONDITIONS:
(GETS X (RETRIEVE2 '%  'PLATFORM))
(GETS Y (RETRIEVE2 'HOSTILITY X))
(GETOP Z THOSTILITY)
(UNLESS (RETRIEVE3 Z 'SAME-AS Y))

ACTION:
(ASSERT Z 'HOSTILITY X)

CONFIDENCE: -1.0

DESCRIPTION:
NIL


NAME: SLOWER-THAN-A-MERCHANT

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE SIGHTING) X))
(GETS W (RETRIEVE2 (QUOTE SPEED) Z))
(RETRIEVE3 W (QUOTE LESS-THAN) 9)

ACTION:
(ASSERT 'MERCHANT 'CATEGORY X)

CONFIDENCE: -.15

DESCRIPTION:
If the speed of a contact is less than 10 knots, than the contact is
unlikely (-.15) to be a merchant.


NAME: FASTER-THAN-A-MERCHANT

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE SIGHTING) X))
(GETS W (RETRIEVE2 (QUOTE SPEED) Z))
(RETRIEVE3 W (QUOTE GREATER-THAN) 25)

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: -.25

DESCRIPTION:
If the speed of a contact is greater than 25 knots, then the contact is
unlikely (-.25) to be a merchant.
```

```
NAME: SPEED-CHANGED

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE SIGHTING) X))
(GETS W (RETRIEVE2 (QUOTE SUCCESSOR) Z))
(GETS Q (RETRIEVE2 (QUOTE SPEED) Z))
(GETS R (RETRIEVE2 (QUOTE SPEED) W))
(UNLESS (RETRIEVE3 Q (QUOTE ROUGHLY-THE-SAME-SPEED-AS) R))

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: -.3

DESCRIPTION:
If a contact changes speed, it's unlikely to be a merchant (-.3).


NAME: COURSE-CHANGED

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE SIGHTING) X))
(GETS W (RETRIEVE2 (QUOTE SUCCESSOR) Z))
(GETS Q (RETRIEVE2 (QUOTE COURSE) Z))
(GETS R (RETRIEVE2 (QUOTE COURSE) W))
(UNLESS (RETRIEVE3 Q (QUOTE ROUGHLY-THE-SAME-COURSE-AS) R))

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: -.3

DESCRIPTION:
If a contact changes course, it's unlikely (-.3) to be a merchant.

NAME: A-DISTANT-POPUP

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE FIRST-SIGHTING) X))
(GETS Q (RETRIEVE2 (QUOTE RANGE) Z))
(RETRIEVE3 Q (QUOTE GREATER-THAN) 30)

ACTION:
(ASSERT 'MERCHANT 'CATEGORY X)

CONFIDENCE: -.2

DESCRIPTION:
If the first sighting of a contact is more than 30 miles from the
home ship, then it's probably not (-.2) a merchant.
```

```
NAME: A-CLOSE-POPUP

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Z (RETRIEVE2 (QUOTE FIRST-SIGHTING) X))
(GETS Q (RETRIEVE2 (QUOTE RANGE) Z))
(RETRIEVE3 Q (QUOTE LESS-THAN) 15)

ACTION:
(ASSERT 'MERCHANT 'CATEGORY X)

CONFIDENCE: -.2

DESCRIPTION:
If the first sighting of a contact is less than 15 miles from the
home ship, then it's probably not (-.2) a merchant.


NAME: INSIDE-A-STORM

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(GETS Z (RETRIEVE2 (QUOTE SIGHTING) X))
(GETS S (RETRIEVE2 (QUOTE % ) (QUOTE STORM)))
(RETRIEVE3 Z (QUOTE INSIDE) S)

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: -.25

DESCRIPTION:
If a sighting of a contact is in a storm, reduce (-.25) the confidence
that the contact is a merchant
```

```
NAME: A-MATCH-FOR-A-KNOWN-MERCHANT

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETOP Y MERCHANTS)
(GETS L2 (RETRIEVE2 (QUOTE LAST-SIGHTING) X))
(GETS L4 (RETRIEVE2 (QUOTE LATITUDE) L2))
(GETS L5 (RETRIEVE2 (QUOTE LONGITUDE) L2))
(GETS L6 (RETRIEVE2 (QUOTE COURSE) L2))
(GETS L7 (RETRIEVE2 (QUOTE SPEED) L2))
(UNLESS (RETRIEVE3 L7 (QUOTE GREATER-THAN) 25))
(UNLESS (RETRIEVE3 L7 (QUOTE LESS-THAN) 9))
(GETS L8 (RETRIEVE2 (QUOTE TIME) L2))
(GETS M2 (RETRIEVE2 (QUOTE LAST-SIGHTING) Y))
(GETS M4 (RETRIEVE2 (QUOTE TIME) M2))
(GETS M5 (RETRIEVE2 (QUOTE LATITUDE) M2))
(GETS M6 (RETRIEVE2 (QUOTE LONGITUDE) M2))
(RETRIEVE3 L6 (QUOTE ROUGHLY-THE-SAME-COURSE-AS) (BEARING (MASSAGE M5)
(MASSAGE M6) (MASSAGE L4) (MASSAGE L5)))
(RETRIEVE3 L7 (QUOTE ROUGHLY-THE-SAME-SPEED-AS) (SPEED (QUOTIENT
(MASSAGE M4) 60.0) (QUOTIENT (MASSAGE L8) 60.0) (DISTANCE (MASSAGE M5)
(MASSAGE M6) (MASSAGE L4) (MASSAGE L5))))

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: .5

DESCRIPTION:
If the course and speed of a contact match the course and speed that a
known merchant would have used to reach the contact, then increase (+.5)
the confidence that the contact is a merchant.


NAME: ID-LANE

CONDITIONS:
(GETOP M MERCHANTLANE)
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(UNLESS (RETRIEVE3 (QUOTE AIR) (QUOTE MEDIUM) X))
(GETS Y (RETRIEVE2 (QUOTE SIGHTING) X))
(RETRIEVE3 Y (QUOTE IN-LANE) M)

ACTION:
(ASSERT Y (QUOTE INSIDE-A-MERCHANTLANE) (QUOTE DUMMY))

CONFIDENCE: 1.0

DESCRIPTION:
If a position of a contact is in a merchantlane, mark it as such.
```

```
NAME: OUTSIDE-ALL-MERCHANTLANES

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE PLATFORM)))
(UNLESS (RETRIEVE3 (QUOTE AIR) (QUOTE MEDIUM) X))
(GETS Y (RETRIEVE2 (QUOTE SIGHTING) X))
(UNLESS (RETRIEVE3 Y (QUOTE INSIDE-A-MERCHANTLANE) (QUOTE DUMMY)))

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: -.2

DESCRIPTION:
If any position of a contact is outside a merchantlane, then decrease
(-.2) the confidence that the contact is a merchant.


NAME: NOT-REACHABLE-BY-ANY-COMBATANT

CONDITIONS:
(GETS X (RETRIEVE2 (QUOTE % ) (QUOTE CONTACTS)))
(GETS Y (RETRIEVE2 (QUOTE FIRST-SIGHTING) X))
(UNLESS (RETRIEVE3 Y (QUOTE REACHABLE-BY-A-COMBATANT) (QUOTE DUMMY)))

ACTION:
(ASSERT (QUOTE MERCHANT) (QUOTE CATEGORY) X)

CONFIDENCE: .45

DESCRIPTION:
If no combatants could have reached the position of a contact, then
the confidence that the contact is a merchant is increased (+.45).


NAME: GOOD-PATH

CONDITIONS:
(RETRIEVE3 FSBLIP 'SIMPLY-WITHIN-REACH SITE)
(UNLESS (RETRIEVE3 FSBLIP 'BLOCKEDFROM SITE))

ACTION:
NIL

CONFIDENCE: 1.0

DESCRIPTION:
If a sighting of a contact is simply within reach of a combatant and
is not blocked form the combatant (say, by a patrol overflight), then
succeed.
```

Appendix E

These are the messages for the demonstration system.

```
((TOR 0 CENTERLAT 65.83 CENTERLON -24.45 VERTICES ((65.3 -27.75)
        (65.99 -28.45)
        (66.88 -27.41)
        (67.11 -24.48)
        (66.5 -21.57)
        (65.46 -20.68)
        (64.67 -22.12)
        (64.67 -25.66)))
 (WORKING-NAME MERCHANT3 SOURCE EXTERNAL TOR 360 CONTENT
                (CATEGORY: MERCHANT SPEED: 10 LATITUDE: 62.97 LONGITUDE:
                        -26.73 TOS: 180 SOURCE: EXTERNAL))
 (WORKING-NAME BLUE SOURCE EXTERNAL TOR 400 CONTENT
                (TOS: 150 LATITUDE: 61.8 LONGITUDE: -20.87 SOURCE: EXTERNAL))
 (WORKING-NAME CONTACT1 LATITUDE 63.34 LONGITUDE -25.52 TOS 420 SOURCE RADAR)
 (WORKING-NAME CONTACT1 LATITUDE 63.37 LONGITUDE -25.41 TOS 440 SOURCE RADAR)
 (WORKING-NAME CONTACT1 LATITUDE 63.4 LONGITUDE -25.31 TOS 460 SOURCE RADAR)
 (WORKING-NAME RED TOR 465 SOURCE EXTERNAL CONTENT
                (LATITUDE: 65.15 LONGITUDE: -28.21 SOURCE: EXTERNAL TOS: 420))
 (WORKING-NAME CONTACT1 LATITUDE 63.43 LONGITUDE -25.21 TOS 480 SOURCE RADAR)
 (WORKING-NAME RED SOURCE EXTERNAL TOR 510 CONTENT
                (TOS: 210 LATITUDE: 65.7 LONGITUDE: -26.61 SOURCE: EXTERNAL))
 (WORKING-NAME RED TOR 580 SOURCE EXTERNAL CONTENT
                (LATITUDE: 66.175 LONGITUDE: -25.2 TOS: 30 SOURCE: EXTERNAL))
 (WORKING-NAME BLUE SOURCE EXTERNAL TOR 630 CONTENT
                (SOURCE: EXTERNAL TOS: 585 CATEGORY: CV LATITUDE: 61.47
                        LONGITUDE: -24.53))
 (WORKING-NAME MERCHANT3 SOURCE PATROL17 TOR 680 CONTENT
                (TOS: 605 LATITUDE: 63.608 LONGITUDE: -24.55 SOURCE: PATROL17
                        CATEGORY: MERCHANT))
 (WORKING-NAME RED SOURCE PATROL17 TOR 685 CONTENT
                (LATITUDE: 63.81 LONGITUDE: -27.95 TOS: 615 SOURCE: PATROL17
                        CLASS: KYNDA))
 (WORKING-NAME CONTACT2 TOS 695 LATITUDE 62.17 LONGITUDE -26.12 SOURCE RADAR)
 (WORKING-NAME CONTACT2 LATITUDE 62.23 LONGITUDE -26.26 TOS 705 SOURCE RADAR)
 (WORKING-NAME CONTACT2 LATITUDE 62.33 LONGITUDE -26.47 TOS 720 SOURCE RADAR)
 (WORKING-NAME CONTACT3 LATITUDE 62.88 LONGITUDE -27.43 TOS 720 SOURCE RADAR)
 (WORKING-NAME CONTACT2 LATITUDE 62.39 LONGITUDE -26.6 TOS 730 SOURCE RADAR)
 (WORKING-NAME BLUE SOURCE EXTERNAL TOR 730 CONTENT
                (LONGITUDE: -23.66 LATITUDE: 61.21 TOS: 510 SOURCE: EXTERNAL))
 (WORKING-NAME CONTACT3 LATITUDE 62.78 LONGITUDE -27.38 TOS 730 SOURCE RADAR)
 (WORKING-NAME CONTACT2 LATITUDE 62.46 LONGITUDE -26.74 TOS 740 SOURCE RADAR)
 (WORKING-NAME CONTACT3 LATITUDE 62.7 LONGITUDE -27.33 TOS 740 SOURCE RADAR)
 (WORKING-NAME PATROL17 SOURCE EXTERNAL TOR 747 CONTENT
                (TOS: 600 LATITUDE: 64.33 LONGITUDE: -22.24 SOURCE: EXTERNAL
                        NAME: HUMMER CLASS: F16-A CATEGORY: FIGHTER))
 (WORKING-NAME PATROL17 SOURCE EXTERNAL TOR 747 CONTENT
                (TOS: 605 LATITUDE: 64.08 LONGITUDE: -24.42 SOURCE: EXTERNAL))
 (WORKING-NAME PATROL17 SOURCE EXTERNAL TOR 747 CONTENT
                (LATITUDE: 63.64 LONGITUDE: -27.93 TOS: 615 SOURCE. EXTERNAL))
```

```
(WORKING-NAME PATROL17 SOURCE EXTERNAL TOR 748 CONTENT
              (LATITUDE: 62.11 LONGITUDE: -36.41 TOS: 640 SOURCE: EXTERNAL)
(WORKING-NAME PATROL17 SOURCE EXTERNAL TOR 748 CONTENT
              (SOURCE: EXTERNAL TOS: 645 LATITUDE: 61.77 LONGITUDE: -37.97)
(WORKING-NAME CONTACT2 LATITUDE 62.52 LONGITUDE -26.89 TOS 750 SOURCE RADAR
(WORKING-NAME CONTACT3 LATITUDE 62.61 LONGITUDE -27.26 TOS 750 SOURCE RADAR
(WORKING-NAME CONTACT2 LATITUDE 62.59 LONGITUDE -27.03 TOS 760 SOURCE RADAR
(WORKING-NAME CONTACT3 LATITUDE 62.53 LONGITUDE -27.2 TOS 760 SOURCE RADAR)
(WORKING-NAME CONTACT2 LATITUDE 62.65 LONGITUDE -27.17 TOS 770 SOURCE RADAR
(WORKING-NAME CONTACT3 LATITUDE 62.44 LONGITUDE -27.14 TOS 770 SOURCE RADAR
(WORKING-NAME BLUE SOURCE EXTERNAL TOR 772 CONTENT
              (SOURCE: EXTERNAL TOS: 330 LATITUDE: 61.5 LONGITUDE: -22.3))
(WORKING-NAME CONTACT2 LATITUDE 62.72 LONGITUDE -27.29 TOS 780 SOURCE RADAR
(WORKING-NAME CONTACT3 LATITUDE 62.35 LONGITUDE -27.07 TOS 780 SOURCE RADAR
```

Appendix F

Annotated Code

NEWSNET
STAMMER
EXPLAIN
SUPP
DSPLA
FORK
WITHINR
EASE

```
(FILECREATED "12-Feb-79 15:02:15" <RBECHTAL>EXPLAIN..53 36407

     changes to:  EXPLAINVARS VARIABLE SPECIALTYPE ST2 ST2B ST3 ST4 ST4B

     previous date: "10-Jan-79 16:49:24" <RBECHTAL>EXPLAIN..52)


(PRETTYCOMPRINT EXPLAINCOMS)

(RPAQQ EXPLAINCOMS [(VARS * EXPLAINVARS)
                    (FNS * EXPLAINFNS)
                    (P (LOAD (QUOTE <PMORRIS>ATN.COM]

(RPAQQ EXPLAINVARS (MISCELLANEOUS VARIABLE SPECIALRELS SPECIALTYPE ST1 ;
                                  ST2 ST2B ST3 ST4 ST4B))

(RPAQQ MISCELLANEOUS ($))

(RPAQQ VARIABLE ($))

(RPAQQ SPECIALRELS (INSIDE-A-MERCHANTLANE REACHABLE-BY-A-COMBATANT))

(RPAQQ SPECIALTYPE (PLATFORM STORM))

(RPAQQ ST1 ("" (((QUOTE (IF AND))
                 ST2
                 (SETQ CNDFLG T)
                 " ")
                ((QUOTE UNLESS)
                 ST2
                 (AND (SETQ CNDFLG T)
                      (SETQ UNLESSFLG T))
                 " ")
                ((QUOTE THEN)
                 ST2
                 (OR (SETQ CNDFLG)
                     (SETQ RTYPE (QUOTE ASSERT)))
                 " "))))

(RPAQQ ST2 ("" (((QUOTE *)                                                    ;
                 ST2B NIL "")                                                 ;
                (MISCELLANEOUS ST3 (SETQ ARG1 (KWOTE KEY))
                               " "))))

(RPAQQ ST2B ("" ((VARIABLE ST3 [AND (SETQ ARG1 KEY)
                                     (COND ((MEMB ARG1 VARS))
                                           (T (SETQ NEWVARS
                                                    (ADD1 NEWVARS))
                                              (SETQ VARS
                                                    (CONS ARG1 VARS))
                                              (SETQ RTYPE (QUOTE
                                                           RETRIEVE2))
                                              (SETQ GETVAR ARG1]
                                ""))))
```

```
(RPAQQ ST3 ("
          (((QUOTE (IS A AN THE))
            ST3 NIL " ")
           (SPECIALTYPE T [COND ((LESSP NEWVARS 1)
                                 (TERPRI)
                                 (PRIN1
                           "ERROR - no variables to bind. Try again.")
                                 (TERPRI))
                                (T (SETQ LNE (LIST (QUOTE GETOP)
                                                   ARG1 KEY]
                      "
")
           (SUBTYPES T [COND ((LESSP NEWVARS 1)
                              (TERPRI)
                              (PRIN1
                           "ERROR - no variables to bind. Try again.")
                              (TERPRI))
                             (T (SETQ LNE (LIST (QUOTE GETOP)
                                                ARG1 KEY]
                    "
")
           (SPECIALRELS
             T
             [COND [(LESSP NEWVARS 1)
                    (COND [UNLESSFLG
                           (SETQ LNE (LIST (QUOTE UNLESS)
                                           (LIST (QUOTE RETRIEVE3)
                                                 ARG1
                                                 (KWOTE KEY)
                                                 (QUOTE (QUOTE DUMMY]
                          (T (SETQ LNE (LIST RTYPE ARG1 (KWOTE KEY)
                                             (QUOTE (QUOTE DUMMY]
                   (T (SETQ LNE (LIST (QUOTE GETS)
                                      GETVAR
                                      (LIST (QUOTE RETRIEVE2)
                                            (KWOTE KEY)
                                            (QUOTE (QUOTE DUMMY]
              "
")
           (RELATIONS ST4 (SETQ REL KEY)
                      " "))))

(RPAQQ ST4 ("" (((QUOTE OF)
                 ST4 NIL " ")
                ((QUOTE *)
                 ST4B NIL "")
                (MISCELLANEOUS T [COND
                                  [(AND UNLESSFLG (LESSP NEWVARS 1))
                                   (SETQ LNE (LIST (QUOTE UNLESS)
                                                   (LIST (QUOTE
                                                           RETRIEVE3)
                                                         ARG1
                                                         (KWOTE REL)
                                                         (KWOTE KEY]
                                  [(LESSP NEWVARS 1)
                                   (SETQ LNE (LIST RTYPE ARG1
```

```
                                                    (KWOTE REL)
                                                    (KWOTE KEY]
                                        (UNLESSFLG (TERPRI)
                                                   (PRIN1
                          "ERROR - unbound variable in UNLESS. Try again.")
                                                   (TERPRI))
                                        (T (SETQ LNE
                                                 (LIST (QUOTE GETS)
                                                       GETVAR
                                                       (LIST (QUOTE RETRIEVE2)
                                                             (KWOTE REL)
                                                             (KWOTE KEY]
                                    "
"))))

(RPAQQ ST4B (""
          ((VARIABLE
             T
             [AND (SETQ ARG2 KEY)
                  (COND [(MEMB ARG2 VARS)
                         (COND [(LESSP NEWVARS 1)
                                (SETQ
                                  LNE
                                  (COND
                                    (UNLESSFLG
                                      (LIST (QUOTE UNLESS)
                                            (LIST (QUOTE RETRIEVE3)
                                                  ARG1
                                                  (KWOTE REL)
                                                  ARG2)))
                                    (T (LIST RTYPE ARG1 (KWOTE REL)
                                             ARG2]
                               (T (SETQ LNE (LIST (QUOTE GETS)
                                                  GETVAR
                                                  (LIST (QUOTE RETRIEVE2)
                                                        (KWOTE REL)
                                                        ARG2]
                        (T (SETQ GETVAR ARG2)
                           (SETQ NEWVARS (ADD1 NEWVARS))
                           (SETQ VARS (CONS ARG2 VARS))
                           (COND ((GREATERP NEWVARS 1)
                                  (TERPRI)
                                  (PRIN1
                          "ERROR - too many unbound variables. Try again.")
                                  (TERPRI))
                                 (T (SETQ LNE
                                          (LIST (QUOTE GETS)
                                                GETVAR
                                                (LIST (QUOTE RETRIEVE2B)
                                                      (KWOTE REL)
                                                      ARG1]
             "
"))))
```



```
(RPAQQ EXPLAINFNS (EXPLAIN HELPEXPLAIN TELLABT WHO1 WHERE1 WHOSE WHY1
                           ISTUFF RESETWHY GOBBLESPEC GETASSRS GETRULES
```

```
                                NEW-LINE NEWRULE NICEATOM NICEASSR NICERULES
                                APDES UNAPDES GAMF GAMF1 PRETTYATOM
                                PRETTYDESCR GETAGOODIE PRINTLI))
(DEFINEQ

(EXPLAIN
  [LAMBDA NIL                                                (* NOBIND
                                                             "10-Jan-79 14:59")
    (PROG ((LASSERS (CONS CONCZ))
           LARULES ACTIVEAS (ASTFLG T))

          (* EXPLAIN is the driver for the explaination
          system. It relies on ASKUSER to make recognition and
          prompting possible.)


          (TERPRI)
          (PRIN1 "Explanation system
Type HELP for help, CR to exit.")
      ELP (TERPRI)
          (COND
            ((EQ [ASKUSER
                   NIL NIL (QUOTE (QUESTION))
                   (QUOTE ((")
" "Leaving EXPLAIN.
" RETURN (PROGN (TEKWAIT)
                 (QUOTE DONE)))
                              (H "elp
" RETURN (HELPEXPLAIN))
                              [E "valuate
" RETURN (PROGN (CLEARBUF)
                 (PRINT (EVAL (READ]
                              (T "ell me about " RETURN (TELLABT))
                              [D "isplay command level
" RETURN (COND
                                   [DSPLAYFLG (COND
                                                (DUALFLG (PRIN1
                                "Enter display commands at the Tektronix.")
                                                  (TERPRI)
                                                  (DSPERASE)
                                                  (DSPTOP))
                                                (T (DSPTOP T]
                                   (T (PRIN1 "no display available
"]
                              (O "K
" RETURN (RESETWHY))
                              (Z "ap
" RETURN (SETQ LASSERS (SETQ LARULES)))
                              (I "s " RETURN (ISTUFF))
                              (WHY " " RETURN
                                   (PROGN (PRIN1 (COND
                                                   (ASTFLG
                                                  " (assertion number) ")
                                                   (T " (rule number) ")))
                                          (WHY1)))
                              (WHOSE " " RETURN (WHOSE))
```

```
                              (WHO " is " RETURN (WHO1))
                              (WHAT " is " RETURN (WHO1))
                              (WHERE " is " RETURN (WHERE1))
                              (N "ew rule definition
" RETURN (NEWRULE]                                                    ;
                    (QUOTE DONE))
                 (RETURN))
               (T (GO ELP)))
             (GO ELP])

(HELPEXPLAIN
  [LAMBDA NIL                                          (* NOBIND
                                                       "30-Nov-78 12:56")
    (PROG (PRMT)
          (PRIN1                                                      ;

"  EXPLAIN is provided to allow you to ask questions about the contents
  of the data base (memory) and how they got there. Question forms are:

     1. WHO is
     2. WHAT is
     3. WHOSE
     4. WHERE is
     5. WHY
     6. Is
     7. Tell me about
     8. Display command level
     9. OK
    10. Zap
    11. Help
    12. Miscellaneous (not a command, but further information)

  Typing a carriage return to a QUESTION ? prompt will cause an exit
  from EXPLAIN.

  For further information about a particular query form, type its number.
  Type 0 to leave Help.
")                                                                    ;
      HE1 (PRIN1 "More about #")
          (SETQ PRMT (RATOM))
          (COND
            ((OR DUALFLG (NOT DSPLAYFLG)))
            (T (DSPERASE)))
          (COND
            ((EQP PRMT 0)                                             ;
              (RETURN))
            ((EQP PRMT 1)
              (PRIN1

"WHO is
     Format: WHO is (THE,A,AN) <RELATION> (OF) <ITEM>
                    [a, an, the, and of are optional]

     Examples: WHO is LESS-THAN 7
               WHO is INSIDE STORM0005
```

```
     Some relations do not require that an item be specified. At
     present, these are INSIDE-A-MERCHANTLANE and REACHABLE-BY-
     A-COMBATANT. For these relations, the command looks like

                    WHO is INSIDE-A-MERCHANTLANE

     For related queries, see WHOSE and WHAT.
")
                  (GO HE1))
                ((EQP PRMT 2)
                  (PRIN1

"WHAT is
     Format: WHAT is (THE,AN,A) <RELATION> (OF) <ITEM>

     Example: WHAT is THE LATITUDE OF POSITION0312

     As with WHO, the, an, a, and of are optional, and an item is
     not required with the relations REACHABLE-BY-A-COMBATANT and
     INSIDE-A-MERCHANTLANE.
")
                  (GO HE1))
                ((EQP PRMT 3)
                  (PRIN1

"WHOSE
    Format: WHOSE <RELATION> is <ITEM>

     WHOSE is roughly the inverse of WHAT, e.g. if WHAT is THE
     POSITION OF SIGHTING0027 is answered with POSITION0026, then
     WHOSE POSITION is POSITION0026 will answer SIGHTING0027.
")
                  (GO HE1))
                ((EQP PRMT 4)
                  (PRIN1

"WHERE is
     Format: WHERE is <OBJECT>

     Example: WHERE is CONTACT7

     Acceptable objects are platforms, merchantlanes, and storms.
")
                  (GO HE1))
                ((EQP PRMT 5)
                  (PRIN1

"WHY
     Format: WHY <NUMBER>

     Example: WHY 3
```



```
     In reply to WHO, WHAT, WHOSE, and some Tell me about questions,
     you will be presented with a numbered list of answers. To follow
     the derivation of any of these, ask WHY followed by the number
     of the answer of interest. You will then be given a list of
```

```
     productions, if the answer was deduced by the system (not taken from
     a message or the technical data base). If such a list appears,
     you can do a WHY to it to view the information which enabled
     the rule. The chain of WHYs may be extended indefinitely, alternating
     between data and rules. See the OK and Zap commands for further
     refinements.
")
               (GO HE1))
             ((EQP PRMT 6)
               (PRIN1

"Is
     Formats: Is (THE,A,AN) <RELATION> (OF) <ITEM> <ITEM>
              Is (THE) <ITEM> (A,AN,THE) <RELATION> (OF) <ITEM>
              Is (THE) <PLATFORM> (A) <ID INFO>

     Examples: Is THE LATITUDE OF POSITION0035 -1.22
               Is RADAR THE SOURCE OF SIGHTING0342
               Is KYNDA2 HOSTILE

     Is checks to see if a given assertion is in the data base. A, an,
     the, and of are optional. Is is fairly flexible in format, to
     allow more natural phrasing. In addition, identification
     information (name, class, category, flag, medium, type, hostility, etc
     is directly available, without the need to give the intervening
     relation.
")
               (GO HE1))
             ((EQP PRMT 7)
               (PRIN1

"Tell me about
     Format: Tell me about <SOMETHING>
             Tell me about <GROUP> <NUMBER>

     Examples: Tell me about MERCHANTS
               Tell me about RULE 5

     Tell me about  is probably the most flexible command. In the first
     format, you may ask about a wide range of things, including
     ITEMs, PRODUCTIONS (RULES), or any subtype (MERCHANTS,
     COMBATANTS, CONTACTS, PATROLS). Using the second format, you
     may examine the details of a particular message or rule (production)
     referred to by number.
")
               (GO HE1))
             ((EQP PRMT 8)
               (PRIN1
```



```
"Display command level
     Format and example: Display command level

     If you have a display available, this pseudo-query will give access
     to the top level of DSPLA, to permit drawing rhumb lines and such
     like. If you are running on a Tektronix in single terminal mode,
     you will have to give the top level DSPLA command G to return to
```

```
     EXPLAIN.
")
               (GO HE1))
              ((EQP PRMT 9)
               (PRIN1

"OK
     Format and example: OK

     OK is a useful part of the WHY feature. WHY functions by maintaining
     a context stack of 'active' answers. OK pops this stack to allow
     you to ask about a different answer in a list you've already asked
     WHY of.
")
               (GO HE1))
              ((EQP PRMT 10)
               (PRIN1

"Zap
     Format and example: Zap

     Zap is the ultimate OK. It clears the WHY context stack completely.
")
               (GO HE1))
              ((EQP PRMT 11)
               (PRIN1

"HELP
     Format and example: HELP

     HELP calls the help function that you're presently in.
")
               (GO HE1))
              ((EQP PRMT 12)
               (PRIN1

"  Information in the data base is represented in the form
     <ITEM> <RELATION> <ITEM>
Most explanation facilities are geared to letting you find out what
is in one of these slots, given fillers of the others. For instance,
WHO and WHAT return the first ITEM in all assertions whose RELATION
and second ITEM are those given. WHOSE returns a second ITEM,
given a RELATION and a first ITEM. Is will
respond to the presence of absence of a fully specified assertion in the
data base.

  EXPLAIN has been designed to make user interface relatively easy. At
most points, typing ? will give a list of possible next entries, and
ESC may be used to complete unambiguous works, or fill in unambiguous
characters. To help the system function properly, please give it time
to do its job. Unacceptable characters are not echoed (the bell rings),
as is the case with excessive typeahead. If asked to [confirm], type a
space. Please end questions with a carriage return.
")
               (GO HE1))
              (T (PRIN1
```

```
      "Sorry, I don't understand what you want.
Please try again.
")
                (GO HE1])

(TELLABT
  [LAMBDA NIL                                          (* NOBIND
                                                       "27-Dec-78 10:15")
    (PROG (CRSR)

          (* TELLABT describes, as well as possible, various
          items including oracles, productions, messages, and
          basenodes.)


      TLP (COND
            [(SETQ CRSR
                (GETAGOODIE (APPEND SUBTYPES ORACLES PRODUCTIONS
                                    BASENODES
                                    (QUOTE (THE ORACLES PRODUCTIONS
                                                MESSAGES MESSAGE
                                                PRODUCTION RULE RULES]
            (T (RETURN)))
          (COND
            ((MEMB CRSR (QUOTE (THE the)))
              (GO TLP))
            ([MEMB CRSR (APPEND SUBTYPES
                                (QUOTE (SUBTYPES TYPES RELATIONS
                                                 ORACLES PRODUCTIONS
                                                 MESSAGES]
              (TERPRI)
              (CLEARBUF)
              (PRINTLI (EVAL CRSR)))
            ((MEMB CRSR ORACLES)
              (TERPRI)
              (CLEARBUF)
              (PRIN1 "It's an oracle, and not printable (compiled)."))
            ((MEMB CRSR PRODUCTIONS)
              (TERPRI)
              (CLEARBUF)
              (FANCYPROD CRSR))
            ((EQ CRSR (QUOTE MESSAGE))
              [SETQ CRSR (CAR (NTH MSGS (RATOM]
              (TERPRI)
              (PRINT CRSR)
              (CLEARBUF))
            ((EQUAL CRSR (QUOTE RULES))
              (TERPRI)
              (CLEARBUF)
              (PRINTLI PRODUCTIONS))
            ((MEMB CRSR (QUOTE (PRODUCTION RULE)))
              [SETQ CRSR (CAR (NTH PRODUCTIONS (RATOM]
              (TERPRI)
              (FANCYPROD CRSR)
              (CLEARBUF))
            (T (TERPRI)
```

```
                (CLEARBUF)
                (NICEASSR (APPEND (GETPROP CRSR (QUOTE TWO*))
                                  (GETPROP CRSR (QUOTE ONE*))
                                  (GETPROP CRSR (QUOTE RELATION*])

(WHO1
  [LAMBDA (FFLG)                                          (* NOBIND
                                                          "27-Dec-78 10:18")
    (PROG (T1)

          (* WHO1 handles WHO and WHAT questions by use of
          GOBBLESPEC. If GOBBLESPEC returns NIL, WHO1 denies
          knowledge.)


          (SETQ T1 (GOBBLESPEC))
          (TERPRI)
          (CLEARBUF)
          (COND
            (T1 (NICEATOM T1))
            (T (PRIN1 "I don't know")
               (TERPRI])

(WHERE1
  [LAMBDA NIL                                             (* NOBIND
                                                          "27-Dec-78 10:37")
    (PROG (PTM PPOS)                                      (* WHERE1 locates
                                                          platforms,
                                                          merchantlanes, and
                                                          storms.)
      WHLP(COND
            [(SETQ PTM (GETAGOODIE (APPEND (QUOTE (THE MERCHANTLANES))
                                           PLATFORM MERCHANTLANE STORM]
            (T (RETURN)))
          (COND
            ((MEMB PTM (QUOTE (THE the)))
              (GO WHLP))
            [(MEMB PTM PLATFORM)
              (TERPRI)
              (CLEARBUF)
              [COND
                ((EQ PTM (QUOTE PERRY))
                  (SETQ PPOS (LIST PERRYLAT PERRYLON)))
                (T (SETQ PPOS (GETPOS (CAAR (RETRIEVE2 (QUOTE
                                                       LAST-SIGHTING)
                                                       PTM]
              (PRIN1 "Last sighted at ")
              (PRIN1 (CAR PPOS))
              (PRIN1 ", ")
              (PRIN1 (CADR PPOS))
              (PRIN1 ".")
              (TERPRI)
              (COND
                (DSPLAYFLG (PRIN1 "Would you like a display? ")
                           (COND
                             ((MEMB (READC)
```

```
                                    (QUOTE (Y y)))
                                (TEKWAIT)
                                (DSPCMD (CONCAT "PTR PERRY," PTM)))
                               (T (TERPRI]
            [(MEMB PTM MERCHANTLANE)
              (TERPRI)
              (CLEARBUF)
              (PRIN1
"The following is a list of lat-lon pairs which lie in the merchant lane."
                     )
              (TERPRI)
              (PRINT (GETVERM PTM))
              (COND
                (DSPLAYFLG (PRIN1
                     "The display will also show the merchantlane.")
                           (TERPRI)
                           (TEKWAIT)
                           (DSPCMD (CONCAT "PTR PL," PTM]
            [(MEMB PTM STORM)
              (TERPRI)
              (CLEARBUF)
              (PRIN1
"The following is a list of lat-lon pairs defining the edge of the storm."
                     )
              (TERPRI)
              (PRINT (GETVERS PTM))
              (COND
                (DSPLAYFLG (PRIN1 "Also, see the display.")
                           (TERPRI)
                           (TEKWAIT)
                           (DSPCMD (CONCAT "PTR PL," PTM]
            [(EQ PTM (QUOTE MERCHANTLANES))
              [MAPC MERCHANTLANE (FUNCTION (LAMBDA (LNM)
                         (PROG NIL
                               (TERPRI)
                               (PRIN1 LNM)
                               (PRIN1
                              " contains the following lat-lon pairs:")
                               (TERPRI)
                               (PRINT (GETVERM LNM))
                               (TERPRI]
              (COND
                (DSPLAYFLG (PRIN1 "Also, see the display.")
                           (TERPRI)
                           (TEKWAIT)
                           (DSPCMD "PTR PL,MLANE"]
            (T (TERPRI)
               (CLEARBUF)
               (PRIN1
                "Expected a platform, merchantlane, or storm name.")
               (TERPRI]))

(WHOSE
  [LAMBDA NIL                                          (* NOBIND
                                                       "27-Dec-78 10:18")
    (PROG (RA VL)                                      (* WHOSE gives a nicer !
```

AD-A116 527 SYSTEM DEVELOPMENT CORP SAN DIEGO CA F/G 17/2
STAMMER: SYSTEM FOR TACTICAL ASSESSMENT OF MULTISOURCE MESSAGES--ETC(U)
MAY 79 R J BECHTEL, P H MORRIS N00123-76-C-0172
UNCLASSIFIED NOSC-TD-252 NL

END
DATE FILMED
07-82
DTIC

```
                                                    way of doing a
                                                    RETRIEVE2B. )
            (COND
              ((SETQ RA (GETAGOODIE RELATIONS)))
              (T (RETURN)))
            (COND
              ((MEMB RA RELATIONS)
                (PRIN1 "is ")
                (COND
                  ((SETQ VL (GETAGOODIE BASENODES)))
                  (T (RETURN)))
                (TERPRI)
                (CLEARBUF)
                (NICEATOM (RETRIEVE2B RA VL)))
              (T (PRIN1 "Expected a relation.")
                 (TERPRI)))

(WHY1
  [LAMBDA NIL                                       (* NOBIND
                                                    "27-Dec-78 10:20")
    (PROG (PTR)

          (* WHY1 provides the mechanism for exploring the
          derivation of an assertion. It reads a number
          (defaults to 1) and uses that number to select an
          assertion or rule of interest.)


          (CLEARBUF)
          (SETQ PTR (CAR (LINEREAD)))
          (COND
            ((NULL PTR)
              (SETQ PTR 1)))
          (CLEARBUF)
          (TERPRI)
          (COND
            (ASTFLG (SETQ LARULES
                          (CONS [GETRULES (SETQ ACTIVEAS
                                                (CAR (NTH (CAR LASSERS)
                                                          PTR]
                                LARULES))
                          (NICERULES (CAR LARULES)))
              (T (GETASSRS ACTIVEAS (CAR (NTH (CAR LARULES)
                                              PTR])

(ISTUFF
  [LAMBDA NIL                                       (* NOBIND
                                                    "27-Dec-78 10:08")
    (PROG (CMPR POSES ANWR)
```



```
          (* ISTUFF handles questions like IS SIGHTING27 A
          SIGHTING OF RED and IS A SIGHTING OF RED SIGHTING27.
          It relies on GOBBLESPEC to return a list of possible
          answers (like all sightings of RED), then compares
          the remaining atom to that list.
          If the item is not in the list, ISTUFF answers
```

```
          "not to my knowledge", otherwise with GAMF1 of the
          appropriate assertion.)


          (COND
            [(SETQ CMPR (GETAGOODIE (APPEND (QUOTE (A AN THE))
                                            BASENODES]
            (T (RETURN)))
          (COND
            [(MEMB CMPR (QUOTE (THE A AN the a an)))
              (SETQ POSES (GOBBLESPEC))
              (COND
                ((SETQ ANWR (ASSOC (GETAGOODIE BASENODES)
                                   POSES))
                  (TERPRI)
                  (CLEARBUF)
                  (GAMF1 (CDR ANWR)))
                (T (TERPRI)
                   (CLEARBUF)
                   (PRIN1 "not to my knowledge")
                   (TERPRI]
            (T (SETQ POSES (GOBBLESPEC))
               (COND
                 ((SETQ ANWR (ASSOC CMPR POSES))
                   (TERPRI)
                   (CLEARBUF)
                   (GAMF1 (CDR ANWR)))
                 (T (TERPRI)
                    (CLEARBUF)
                    (PRIN1 "not to my knowledge")
                    (TERPRI])

(RESETWHY
  [LAMBDA NIL                                          (* NOBIND
                                                       "27-Dec-78 10:14")

          (* RESETWHY changes the ASTFLG and pops the
          appropriate WHY stack (either assertions or rules)
          to back up in a WHY descent.)


    [COND
      (ASTFLG (SETQ LASSERS (CDR LASSERS)))
      (T (SETQ LARULES (CDR LARULES]
    (SETQ ASTFLG (NOT ASTFLG])

(GOBBLESPEC
  [LAMBDA NIL                                          (* NOBIND
                                                       "27-Dec-78 10:04")
    (PROG (NXT)
```



```
          (* GOBBLESPEC expects to see a phrase of the form
          <a, an, the> RELATION <of> <a, an, the> BASENODE or
          <a, an, the> PLATFORM or INSIDE-A-MERCHANTLANE,
          REACHABLE-BY-A-COMBATANT or <id info>, then does a
          RETRIEVE2 as appropriate, and returns that as the
```

```
answer. At present, GOBBLESPEC can only deal with a
single relation.)


GSLP(COND
      [(SETQ NXT (GETAGOODIE (APPEND (QUOTE (THE AN A))
                                     TNAME TTYPE THOSTILITY
                                     TAGGRESSIVENESS TCATEGORY
                                     TCLASS TFLAG TMEDIUM
                                     PLATFORM RELATIONS]
       (T (RETURN)))
     [COND
       ((MEMB NXT (QUOTE (THE AN A the an a)))
         (GO GSLP))
       [(MEMB NXT (QUOTE (INSIDE-A-MERCHANTLANE
                                     REACHABLE-BY-A-COMBATANT)))
         (RETURN (RETRIEVE2 NXT (QUOTE DUMMY]
       [(MEMB NXT RELATIONS)
         (RETURN (PROG (AFTR)
                       IGSLP
                            (COND
                              [(SETQ AFTR
                                   (GETAGOODIE (APPEND (QUOTE (OF))
                                                       BASENODES]
                              (T (RETURN)))
                            (COND
                              ((MEMB AFTR (QUOTE (OF of)))
                                (GO IGSLP))
                              (T (RETURN (RETRIEVE2 NXT AFTR]
       ((MEMB NXT PLATFORM)                    (* This handles WHAT IS
                                                  RED, etc.)
         (RETURN (PROG (ANNS)
                       (COND
                         ((OR (SETQ ANNS (RETRIEVE2 (QUOTE NAME)
                                                    NXT))
                              (SETQ ANNS (RETRIEVE2 (QUOTE CLASS)
                                                    NXT))
                              (SETQ ANNS (RETRIEVE2 (QUOTE
                                                      CATEGORY)
                                                    NXT))
                              (SETQ ANNS (RETRIEVE2 (QUOTE
                                                     HOSTILITY)
                                                    NXT))
                              (SETQ ANNS (RETRIEVE2 (QUOTE TYPE)
                                                    NXT))
                              (SETQ ANNS (RETRIEVE2 (QUOTE MEDIUM)
                                                    NXT)))
                          (RETURN ANNS]    (* The remaining
                                              possibilities concern id
                                              information, e.g. IS THE
                                              PERRY HOSTILE)
     (COND
       ((MEMB NXT TAGGRESSIVENESS)
         (RETURN (RETRIEVE2B (QUOTE WARLIKE)
                             NXT)))
       ((MEMB NXT THOSTILITY)
```

```
                (RETURN (RETRIEVE2B (QUOTE HOSTILITY)
                                    NXT)))
              ((MEMB NXT TMEDIUM)
                (RETURN (RETRIEVE2B (QUOTE MEDIUM)
                                    NXT)))
              ((MEMB NXT TFLAG)
                (RETURN (RETRIEVE2B (QUOTE FLAG)
                                    NXT)))
              ((MEMB NXT TTYPE)
                (RETURN (RETRIEVE2B (QUOTE TYPE)
                                    NXT)))
              ((MEMB NXT TNAME)
                (RETURN (RETRIEVE2B (QUOTE NAME)
                                    NXT)))
              ((MEMB NXT TCLASS)
                (RETURN (RETRIEVE2B (QUOTE CLASS)
                                    NXT)))
              ((MEMB NXT TCATEGORY)
                (RETURN (RETRIEVE2B (QUOTE CATEGORY)
                                    NXT])

(GETASSRS
  [LAMBDA (AS RL)                                    (* NOBIND
                                                     "27-Dec-78 09:31")
    (PROG (ACTBOX BOXLST)

          (* GETASSRS collects the assertions needed to
          respond to a WHY question. Since a particular rule
          may have concluded the same thing in several ways,
          the user is given the opportunity to ask about each
          set of assertions separately.
          Only if the user wishes to ask about an assertion is
          ASTFLG reset, otherwise they may then ask about
          other rules.)


          (SETQ BOXLST (GETPROP AS (QUOTE DERIVE*)))
      GA1 (COND
            ((NULL BOXLST)
              (RETURN))
            (T (SETQ ACTBOX (CAR BOXLST))
               (SETQ BOXLST (CDR BOXLST))
               [COND
                 ([MEMB RL (SETQ ALI (GETPROP ACTBOX (QUOTE FROM]
                   [NICEASSR (APPEND (RELDIF ALI PRODUCTIONS)
                                     (GETPROP ACTBOX (QUOTE NEGFROM]
                   (TERPRI)
                   (PRIN1 "Do you want to know about any of these? ")
                   (SELECTQ (ASKUSER)
                            ((Y y)
                              (CLEARBUF)
                              (SETQ ASTFLG T)
                              (PRIN1
                          "Ask WHY about them at the next QUESTION.
                                    98
                              (RETURN))
```

```
                                (PROGN (TERPRI)
                                       (CLEARBUF)
                                       (SETQ LASSERS (CDR LASSERS))
                                       (SETQ ASTFLQ NIL)
                                       (GO GA1])
                        (GO GA1])

(GETRULES
  [LAMBDA (ASSRST)                                        (* NOBIND
                                                          "27-Dec-78 09:32")
    (PROG (Q1 Q2)

          (* GETRULES collects the rules needed to answer a
          WHY question. Printing is done in NICERULE.)


          (SETQ Q1 (GETPROP ASSRST (QUOTE DERIVE*)))
          (COND
            ((NULL Q1)
              (RETURN)))
      GR1 (COND
            ((NULL Q1)
              (RETURN (INTERSECTION PRODUCTIONS Q2)))
            (T (SETQ Q2 (APPEND (GETPROP (CAR Q1)
                                         (QUOTE FROM))
                                Q2))
               (SETQ Q1 (CDR Q1))
               (GO GR1])

(NEW-LINE
  [LAMBDA (OLDVARS)
    (PROG (VARS ARG1 ARG2 GETVAR REL (NEWVARS 0)
                (RTYPE (QUOTE RETRIEVE3))
                LNE CNDFLQ UNLESSFLQ)
          (SETQ VARS OLDVARS)
          (USERQUERY ST1)
          (RETURN (LIST CNDFLQ VARS LNE])

(NEWRULE
  [LAMBDA NIL                                             (* NOBIND
                                                          "10-Jan-79 16:15")
    (PROG (NE CNDS ATS CONF LAAT OV)
          (CONTROL T)
          (TERPRI)
          (PRIN1 "Rule name? ")
          (SETQ NE (RATOM))
          (TERPRI)
          (PRIN1 "Text?")
          (TERPRI)
      CONDLOOP
          (CLEARBUF)
          (PRIN1 ": ")
          (SETQ LAAT (NEW-LINE OV))
          (COND
            ((NULL (CAR LAAT))
              (SETQ ATS (CADDR LAAT))
```

```
                (GO OUTERLP))
               (T (SETQ OV (CADR LAAT))
                  (SETQ CNDS (APPEND CNDS (CDDR LAAT)))
                  (GO CONDLOOP)))
        OUTERLP
             (TERPRI)
             (PRIN1 "Confidence? ")
             (SETQ CONF (RATOM))
             (TERPRI)
             (MAKEPD NE CNDS ATS CONF)
             (TERPRI)
             (PUTPROP NE (QUOTE WHY)
                      "  because of a user rule.")
             (SETQ CONTEXT)
             (COND
               ((APPLYRULE NE)
                 (PRIN1
                   "This rule has an effect in the existing database.")
                 (TERPRI))
               (T (PRIN1
     "This rule has no effect on the database in it's present form.")
                  (TERPRI)))
             (CONTROL)
             (RETURN NE])

(NICEATOM
  [LAMBDA (LATS)                                    (* NOBIND
                                                    "27-Dec-78 10:09")
    (PROG ((ATCT 1))

          (* NICEATOM updates the WHY assertion list and
          prints a list of modified (GAMFed) atoms.)


          (SETQ LASSERS (CONS (MAPCAR LATS (FUNCTION CDR))
                              LASSERS))
          (SETQ ASTFLG T)
      NAT1(COND
            ((NULL LATS)
              (RETURN))
            (T (PRIN1 ATCT)
               (PRIN1 ". ")
               (PRETTYATOM (CAR LATS))
               (SETQ ATCT (ADD1 ATCT))
               (SETQ LATS (CDR LATS))
               (GO NAT1])

(NICEASSR
  [LAMBDA (LISTASSERS)                              (* NOBIND
                                                    "27-Dec-78 10:08")
    (PROG ((ASCT 1))                                (* NICEASSR updates the
                                                    WHY assertion list and
                                                    prints a list of
                                                    assertions)
          (TERPRI)
          (SETQ LASSERS (CONS LISTASSERS LASSERS))
```

```
      NA1 (COND
            ((NULL LISTASSERS)
              (RETURN))
            (T (PRIN1 ASCT)
               (PRIN1 ". ")
               (PRETTYDESCR (CAR LISTASSERS))
               (SETQ ASCT (ADD1 ASCT))
               (SETQ LISTASSERS (CDR LISTASSERS))
               (GO NA1])

(NICERULES
  [LAMBDA (RULEL)                                        (* NOBIND
                                                         "27-Dec-78 10:36")
    (PROG NIL*

          (* NICERULES updates the WHY rule list and prints a
          list of rule names. If no rules were used in a
          derivation, it so states, and DOESNOT reset ASTFLG.)


          (TERPRI)
          [COND
            ((NULL RULEL)
              (COND
                ((MEMB (CAR (GETPROP ACTIVEAS (QUOTE RELATION)))
                       ORACLES)
                  (PRIN1 "The result of an oracle.")
                  (TERPRI))
                (T (PRIN1
" Either taken directly from a message or part of the existing data base."
                           )
                   (TERPRI)
                   (RETURN]
          (SETQ ASTFLG NIL)
          (PRINTLI RULEL])

(APDES
  [LAMBDA (ATOMM)                                        (* NOBIND
                                                         "27-Dec-78 09:25")

          (* APDES creates an "apt description" for nodes in
          the network. Number nodes are evaluated, sightings,
          vertices, and storm nodes are printed as such, and
          CONTACTS, etc. are printed without the concluding
          s.)


    (COND
      ((MEMB ATOMM NUMNUMS)
        (EVAL ATOMM))
      [(MEMB ATOMM SIGHTINGS)
        (PACK (APPEND (QUOTE (S I G H T I N G))
                      (CDR (UNPACK ATOMM]
      [(MEMB ATOMM POSITION)
        (PACK (APPEND (QUOTE (P O S I T I O N))
                      (CDR (UNPACK ATOMM]
```

```
      [(MEMB ATOMM VERTEX)
        (PACK (APPEND (QUOTE (V E R T E X))
                      (CDR (UNPACK ATOMM]
      [(MEMB ATOMM STORM)
        (PACK (APPEND (QUOTE (S T O R M))
                      (CDR (UNPACK ATOMM]
      [(MEMB ATOMM SUBTYPES)
        (PACK (REVERSE (CDR (REVERSE (UNPACK ATOMM]
      ((EQ ATOMM (QUOTE DUMMY))
        " ")
      (T ATOMM]))

(UNAPDES
  [LAMBDA (FATM)                                        (* NOBIND
                                                        "27-Dec-78 10:16")
    (PROG (PLACE)                                       (* UNAPDES takes an
                                                        "apt description" and
                                                        returns a network node.

          (COND
            ((NUMBERP FATM)
              (RETURN (GETADUMMYFOR FATM)))
            ((MEMB FATM (QUOTE (SIGHTING POSITION VERTEX STORM)))
              (RETURN FATM))
            [(SETQ PLACE (OR (STRPOS "SIGHTING" FATM NIL NIL T T)
                             (STRPOS "POSITION" FATM NIL NIL T T)
                             (STRPOS "VERTEX" FATM NIL NIL T T)))
              (RETURN (MKATOM (CONCAT "N" (SUBSTRING FATM PLACE]
            [(SETQ PLACE (STRPOS "STORM" FATM NIL NIL T T))
              (RETURN (MKATOM (CONCAT "S" (SUBSTRING FATM PLACE]
            (T (RETURN FATM]))

(GAMF
  [LAMBDA (WLK)                                         (* NOBIND
                                                        "27-Dec-78 09:27")
    (PROG (CONFI ACON)

          (* GAMF generates an appropriate modifier for an
          assertion based on the confidence of the assertion.)


          (SETQ CONFI (GETCON WLK))
          (SETQ ACON (ABS CONFI))
          (COND
            ((EQP ACON 1.0))
            ((FGREATERP ACON .98)
              (PRIN1 "definitely "))
            ((FGREATERP ACON .9)
              (PRIN1 "almost certainly "))
            ((FGREATERP ACON .7)
              (PRIN1 "very probably "))
            ((FGREATERP ACON .45)
              (PRIN1 "probably "))
            ((EQP ACON 0.0)
              (PRIN1 "not known to be ")
              (RETURN))
            (T (PRIN1 "somewhat ")
```

```
                (COND
                  ((FLESSP CONFI 0.0)
                    (PRIN1 "un")))
                (PRIN1 "likely to be ")
                (RETURN)))
            (COND
              ((FLESSP CONFI 0.0)
                (PRIN1 "not "])

(GAMF1
  [LAMBDA (ASTR)                                    (* NOBIND
                                                    "27-Dec-78 09:28")
    (PROG (CF)

          (* GAMF1 is used to generate yes-no type answers,
          based on the confidence of an assertion.)


          (SETQ CF (GETCON ASTR))
          (COND
            ((EQP CF 1.0)
              (PRIN1 "yes"))
            ((EQP CF 0.0)
              (PRIN1 "not to my knowledge"))
            ((EQP CF -1.0)
              (PRIN1 "no"))
            (T (GAMF ASTR)))
          (TERPRI])

(PRETTYATOM
  [LAMBDA (PAIR)                                    (* NOBIND
                                                    "27-Dec-78 10:11")
    (PROG NIL

          (* PRETTYATOM prettyprints an atom using GAMF and
          APDES. It expects a dotted pair -
          atom . assertion.)


          (GAMF (CDR PAIR))
          (PRIN1 (APDES (CAR PAIR)))
          (TERPRI])

(PRETTYDESCR
  [LAMBDA (ASR)                                     (* NOBIND
                                                    "27-Dec-78 10:12")
    (PROG (LA TE DA)                                (* PRETTYDESCR is an
                                                    assertion
                                                    prettyprinter.)
          [SETQ LA (CAR (FIND1 (LIST (QUOTE ONE*)
                                     ASR]
          [SETQ TE (CAR (FIND1 (LIST (QUOTE RELATION*)
                                     ASR]
          [SETQ DA (CAR (FIND1 (LIST (QUOTE TWO*)
                                     ASR]
          (COND
```

```
        ((OR (NULL LA)
             (NULL TE)
             (NULL DA))
          (PRIN1 "Information no longer available.")
          (TERPRI)
          (RETURN)))
       (PRIN1 (APDES LA))
       (PRIN1 " is ")
       (GAMF ASR)
       (COND
         [(OR (ORACLE TE)
              (EQ DA (QUOTE DUMMY]
         ([OR (MEMB TE (QUOTE (SIGHTING VERTEX)))
              (MEMB DA SUBTYPES)
              (MEMB DA
                    (QUOTE (PLATFORM STORM MERCHANTLANE TTYPE
                                     TMEDIUM TAGGRESSIVENESS
                                     TCATEGORY TCLASS THOSTILITY
                                     TFLAG TNAME]
          (PRIN1 "a "))
         (T (PRIN1 "the ")))
       (PRIN1 TE)
       (COND
         ([OR (ORACLE TE)
              (EQ DA (QUOTE DUMMY))
              (MEMB DA SUBTYPES)
              (MEMB DA
                    (QUOTE (PLATFORM STORM MERCHANTLANE TTYPE
                                     TMEDIUM TAGGRESSIVENESS
                                     TCATEGORY TCLASS THOSTILITY
                                     TFLAG TNAME]
          (PRIN1 " "))
         (T (PRIN1 " of ")))
       (PRIN1 (APDES DA))
       (TERPRI])

(GETAGOODIE
  [LAMBDA (LISTA)                                    (* NOBIND
                                                     "27-Dec-78 10:26")
    (PROG (POSIBL POS2)

          (* GETAGOODIE provides a means of calling ASKUSER
          with a keylist to be evaaluated.
          GETAGOODIE also provides for spelling correction.)


          (SETQ POSIBL (ASKUSER
              NIL NIL NIL [APPEND LISTA
                                  (QUOTE (("" NIL RETURN
                                           (PROGN (SETQ POS2
                                                    (RATOM))
                                                  NIL]
              T))
          (COND
            (POSIBL (RETURN (UNAPDES POSIBL)))
            (T (COND
```

```
                    ((SETQ POSIBL (MISSPELLED? POS2 70 LISTA))
                      (RETURN POSIBL))
                    (T (TERPRI)
                       (PRIN1 "I have no knowledge of ")
                       (PRIN1 POS2)
                       (TERPRI)
                       (RETURN])

(PRINTLI
  [LAMBDA (XLI)                                          (* NOBIND
                                                         "27-Dec-78 10:13")
    (PROG ((PCT 1))                                      (* PRINTLI prints a
                                                         list, numbering as it
                                                         goes.)

      PRLP(COND
             ((NULL XLI)
               (RETURN))
             (T (PRIN1 PCT)
                (PRIN1 ". ")
                (PRIN1 (CAR XLI))
                (TERPRI)
                (SETQ PCT (ADD1 PCT))
                (SETQ XLI (CDR XLI))
                (GO PRLP])
)
(LOAD (QUOTE <PMORRIS>ATN.COM))
(DECLARE: DONTCOPY
  (FILEMAP (NIL (4686 36350 (EXPLAIN 4698 . 6264) (HELPEXPLAIN 6268 .
13018) (TELLABT 13022 . 14631) (WHO1 14635 . 15145) (WHERE1 15149 .
17476) (WHOSE 17480 . 18054) (WHY1 18058 . 18886) (ISTUFF 18890 . 20218)
 (RESETWHY 20222 . 20680) (GOBBLESPEC 20684 . 24024) (GETASSRS 24028 .
25416) (GETRULES 25420 . 26043) (NEW-LINE 26047 . 26282) (NEWRULE 26286
. 27927) (NICEATOM 27931 . 28564) (NICEASSR 28568 . 29141) (NICERULES
29145 . 30017) (APDES 30021 . 31042) (UNAPDES 31046 . 31786) (GAMF 31790
 . 32746) (GAMF1 32750 . 33295) (PRETTYATOM 33299 . 33724) (PRETTYDESCR
33728 . 35088) (GETAGOODIE 35092 . 35897) (PRINTLI 35901 . 36347)))))
STOP
```

```
(FILECREATED "18-Dec-78 14:27:28" <PMORRIS>ATN.LSP.26 3665

     changes to:  ADDQKLST ATNCOMS USERQUERY

     previous date: "15-Dec-78 18:56:50" <PMORRIS>ATN.LSP.23)


(PRETTYCOMPRINT ATNCOMS)

(RPAQQ ATNCOMS [(FNS * ATNFNS)
                [P (SETQ QSCRATCHPTR (SETQQ QUERYSCRATCHLST (?]
                (VARS (QSCRATCHUNUSED NIL))
                (BLOCKS (ATNBLOCK ADDQKLST MKQKLST USERQUERY
                                  (SPECVARS KEY)
                                  (LOCALFREEVARS PTR)
                                  (GLOBALVARS QUERYSCRATCHLST
```

```
                                                    QSCRATCHPTR
                                                    QSCRATCHUNUSED)
                                          (ENTRIES USERQUERY)
                                          (NOLINKFNS . T])

(RPAQQ ATNFNS (ADDQKLST MKQKLST USERQUERY))
(DEFINEQ

(ADDQKLST
  [LAMBDA (K)                                              (* NOBIND
                                                           "18-Dec-78 12:06")
    (PROG ((CPTR (CDR PTR)))
          [COND
            (CPTR (RPLACA CPTR K))
            (T (RPLACD PTR (CONS K]
          (SETQ PTR (CDR PTR])

(MKQKLST
  [LAMBDA (QLST)                                           (* NOBIND
                                                           "14-Dec-78 13:05")
    (PROG ((PTR QUERYSCRATCHLST))
          (NCONC QSCRATCHPTR QSCRATCHUNUSED)
          [MAPC QLST (FUNCTION (LAMBDA (Q)
                     (PROG [(KEYS (EVAL (CAR Q]
                           (COND
                             ((LISTP KEYS)
                               (MAPC KEYS (FUNCTION ADDQKLST)))
                             (T (ADDQKLST KEYS]
          (SETQ QSCRATCHUNUSED (CDR PTR))
          (SETQ QSCRATCHPTR (RPLACD PTR))
          (RETURN QUERYSCRATCHLST])

(USERQUERY
  [LAMBDA (ISTATE)                                         (* NOBIND
                                                           "18-Dec-78 14:26")
    (PROG ((PRINTBUF "")
           (STATE ISTATE)
           KEYLST KEY MATCHED QUERYLST NEWSTATE)
      LOOP(COND
            ((NULL STATE)
              (RETURN))
            ((EQ STATE T)
              (RETURN)))
          (SETQ QUERYLST (CADR STATE))
          (SETQ KEYLST (MKQKLST QUERYLST))
      ASK [SETQ KEY (RESETFORM (CONTROL T)
                               (ASKUSER NIL NIL (OR (CAR STATE)
                                                    "")
                                        KEYLST T NIL
                                        (QUOTE (CONFIRMFLG NIL]
          (CLEARBUF)
          (COND
            ((EQ KEY (QUOTE ?))
              (TERPRI)
              (TERPRI)
              (PRIN1 "one of:")
```

```
            (TERPRI)
            [MAPC QUERYLST
                  (FUNCTION (LAMBDA (ALT)
                      (PROG ((KEYTYPE (CAR ALT))
                             EK)
                            (SETQ EK (EVAL KEYTYPE))
                            (COND
                              [(EQ (CAR KEYTYPE)
                                   (QUOTE QUOTE))
                                (COND
                                  ((LISTP EK)
                                    (MAPC EK (FUNCTION PRINT)))
                                  (T (PRINT EK]
                              ((LISTP EK)
                                (PRIN1 "a ")
                                (PRINT KEYTYPE))
                              (T (PRINT (MKATOM EK]
            (TERPRI)
            (PRIN1 PRINTBUF)
            (GO ASK)))
          [SETQ MATCHED
            (CAR
              (SOME QUERYLST
                    (FUNCTION (LAMBDA (ALT)
                        (PROG [(ECA (EVAL (CAR ALT]
                              (RETURN (COND
                                        [(LISTP ECA)
                                          (OR (MEMB KEY ECA)
                                              (EQ (CAR ECA)
                                                  (CHARACTER 27]
                                        (T (EQ KEY (MKATOM ECA]
          (PRIN1 (OR (CADDDR MATCHED)
                     ""))
          (EVAL (CADDR MATCHED))
          (SETQ PRINTBUF (CONCAT PRINTBUF (CAR STATE)
                                 KEY
                                 (OR (CADDDR MATCHED)
                                     ""])))
          (SETQ NEWSTATE (EVAL (CADR MATCHED)))
          [SETQ STATE (COND
            ((NUMBERP NEWSTATE)
              (NTH STATE (IPLUS NEWSTATE NEWSTATE 3)))
            (NEWSTATE)
            (T (CDDR STATE]
          (GO LOOP])
)
(SETQ QSCRATCHPTR (SETQQ QUERYSCRATCHLST (?)))

(RPAQ QSCRATCHUNUSED NIL)
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: ATNBLOCK ADDQKLST MKQKLST USERQUERY (SPECVARS KEY)
        (LOCALFREEVARS PTR)
        (GLOBALVARS QUERYSCRATCHLST QSCRATCHPTR QSCRATCHUNUSED)
        (ENTRIES USERQUERY)
        (NOLINKFNS . T))
```

```
(DECLARE: DONTCOPY
  (FILEMAP (NIL (608 3331 (ADDQKLST 620 . 886) (MKQKLST 890 . 1388) (
USERQUERY 1392 . 3328)))))
STOP
```

INITIAL DISTRIBUTION

NAVAL ELECTRONIC SYSTEMS COMMAND
CODE 330 (C. C. STOUT) (2)

NAVAL RESEARCH LABORATORY
CODE 7509 (J. H. KULBACK)
CODE 7932 (H. L. WIENER)

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
IPTO (COL D. C. RUSSELL) (2)
IPTO (LCDR A. J. DIETZLER) (2)

SYSTEMS DEVELOPMENT CORP.
3065 ROSECRANS PLACE
SAN DIEGO, CA 92110
R. J. BECHTEL (4)
P. H. MORRIS (4)

MITRE CORP.
WESTGATE RESEARCH PARK
MC LEAN, VA 22101
J. W. BENOIT

RAND CORP.
1700 MAIN STREET
SANTA MONICA, CA 90406
G. MARTINS

BOLT, BERANEK AND NEWMAN
50 MOULTON STREET
CAMBRIDGE, MA 02138
R. J. BOBROW
N. R. GREENFIELD


DAT
FILM